APPLICATIONS OF MONODROMY IN SOLVING POLYNOMIAL SYSTEMS

A Dissertation
Presented to
The Academic Faculty

By

Timothy Duff

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in Algorithms, Combinatorics, and Optimization
Home unit: School of Mathematics

Georgia Institute of Technology

August 2021

APPLICATIONS OF MONODROMY IN SOLVING POLYNOMIAL SYSTEMS

Thesis committee:

Dr. Anton Leykin (chair)
School of Mathematics
Georgia Institute of Technology

Dr. Rekha Thomas (reader)
Department of Mathematics
University of Washington

Dr. Josephine Yu
School of Mathematics
Georgia Institute of Technology

Dr. Matthew Baker
School of Mathematics
Georgia Institute of Technology

Dr. Gregory Blekherman
School of Mathematics
Georgia Institute of Technology

Dr. Richard Peng
School of Computer Science
Georgia Institute of Technology

Date approved: June 16 2021

ACKNOWLEDGMENTS

First of all, I thank Anton Leykin for teaching me so much, and for always supporting and encouraging me. I couldn't have asked for a better advisor. I am grateful to all of my many collaborators whose shared work appears at various points in this thesis. I am particularly grateful to Kathlén Kohn, Tomas Pajdla, and Frank Sottile for many inspiring conversations that contributed to the results and point of view of this thesis, and to Viktor Korotynskiy and Maggie Regan who helped greatly in bringing Chapter 4 to life.

Thanks to my amazing teachers, mentors, and peers throughout life—you know who you are! I am grateful to Matt Baker, Greg Blekherman, Richard Peng, Rekha Thomas, and Josephine Yu for serving on my committee and for the guidance they've provided. Thanks to John Voight for suggesting the example involving M_{23}. I thank the mathematical communities that have included me—the ACO program, Macaulay2, "nonlinear algebra", ... Thanks to Jose Acevedo, Michael Burr, Marcel Celaya, Ollie Clarke, Trevor Gunn, Cvetelina Hill, Jaewoo Jung, Adrián Pérez Bustamante, Jose Rodriguez, Michael Ruddy, Simon Telen, Jeff Sommars, Elise Walker, James Wenk, Thomas Yahl, fellow "Ley kin"—Justin Chen, Marc Härkönen, Kisun Lee, Robert Krone, and so many others for friendly and insightful discussions that helped me keep the ball rolling.

And thanks to non-math friends like Robert, Laurel, Julia, Blair, ..., for helping me keep my head on straight, and to Mom, Dad, Emma, and Colby for being kind to me even when it wasn't.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

Polynomial systems of equations that occur in applications frequently have a special structure. Part of that structure can be captured by an associated Galois/monodromy group. This makes numerical homotopy continuation methods that exploit this monodromy action an attractive choice for solving these systems; by contrast, other symbolic-numeric techniques do not generally see this structure. Naturally, there are trade-offs when monodromy is chosen over other methods. Nevertheless, there is a growing literature demonstrating that the trade can be worthwhile in practice.

In this thesis, we consider a framework for efficient monodromy computation which rivals the state-of-the-art in homotopy continuation methods. We show how its implementation in the package `MonodromySolver` can be used to efficiently solve challenging systems of polynomial equations. Among many applications, we apply monodromy to computer vision—specifically, the study and classification of minimal problems used in RANSAC-based 3D reconstruction pipelines. As a byproduct of numerically computing their Galois/monodromy groups, we observe that several of these problems have a decomposition into algebraic subproblems. Although precise knowledge of such a decomposition is hard to obtain in general, we determine it in some novel cases.

CHAPTER 1
INTRODUCTION

1.1 Families of polynomial systems

Scientists, engineers, and mathematicians alike are frequently confronted with the need to solve polynomial systems of equations. It is rather typical that these systems occur naturally in a *parametric family:*

$$\begin{aligned} f(x;z) = 0, \text{ where} \\ x = (x_1, \ldots, x_n) \text{ are } \textit{variables}, \\ z = (z_1, \ldots, z_m) \text{ are } \textit{parameters}, \\ f = (f_1, \ldots, f_N) \text{ are } \textit{polynomials.} \end{aligned} \tag{1.1}$$

The *parameters* z represent given measurements or data. The *variables* are unknown quantities to be solved for. The system of equations $f_1(x;z) = \cdots = f_N(x;z) = 0$ may describe a mathematical object or encode constraints implied by some underlying model. Methods for solving these systems of equations lie within the intersection of algebraic geometry and applied mathematics, a subject that has been dubbed *nonlinear algebra* [86, 22].

It is well-known that solving polynomial systems, and algebraic geometry in general, works best over the complex numbers. This is due in large part to classical principles, such as *dimension counting* and *conservation of number.* It is a standard trick in algebraic geometry to consider the entire family of systems in Equation 1.1 at once as a geometric object, by considering the incidence correspondence of parameter-solution pairs

$$X = \{(x, z) \in \mathbb{C}^n \times \mathbb{C}^m \mid f(x;z) = 0\}. \tag{1.2}$$

Let us first note that the set X is itself the set of solutions to a polynomial system of equations. This is a *feature* of algebraic geometry that distinguishes it from its mathematical

cousins, and a compelling reason to search for algebraic models wherever they might occur. Let us also note that the solution sets which interest us, namely $\{x \in \mathbb{C}^n \mid f(x; z) = 0\}$, may be naturally identified with the fibers $X_z := \{(x, z) \in \mathbb{C}^n \mid f(x; z) = 0\}$ of the map which projects X onto the space of parameters:

$$\begin{aligned} X &\to \mathbb{C}^m \\ (x, z) &\mapsto z. \end{aligned} \tag{1.3}$$

One would like to invert this map. There are three possibilities for a generic fiber X_z:

- X_z is empty (*an over-constrained problem*) ,
- $\dim X_z > 0$ (*an under-constrained problem*) , or
- there are finitely many solutions: $X_z = \{x_1, \ldots, x_d\} \times \{z\}$, with d independent of z.

When we are lucky enough to be in the third, *well-constrained* case, the number of solutions d may still be prohibitively large for the intended application. If our goal is to compute all solutions, working over the real numbers does not help us much in general. Indeed, the cardinality of a typical real fiber $X_z \cap \mathbb{R}^n$ is not conserved (consider $X = \{(x, z) \mid x^2 - z = 0\}$.) Moreover, in applications it often holds that X is an *irreducible* variety with a smooth real point. This implies that the real points $X(\mathbb{R})$ are *Zariski-dense* in X, and hence the complex solution set X_z will have the same cardinality for generic z, real or complex.

It is reasonable to take a more optimistic stance than in the previous paragraph. For one, there are many natural applications where solving a well-constrained system is the immediate goal. One rather prominent example comes from the study of *minimal problems*, which are used inside of RANSAC-based pipelines in 3D reconstruction. A systematic study of these problems is presented in Chapter 3 of this thesis. Other natural examples include problems where we must compute the equillibria of dynamical systems, or the assembly modes of mechanisms in kinematics, or enumerative problems in geometry.

In other applications, a well-constrained system may serve as a proxy for the real problem of interest. For instance, an over-constrained problem may be recast as an optimization problem—we may then reduce to the well-constrained case by considering the associated *critical point systems.* At the opposite end of the spectrum, positive-dimensional solution sets can be approached via their intersections with generic linear spaces—this is the main principle underlying the algorithms of *numerical algebraic geometry.* Thus, the interesting cases where our system of equations typically has finitely many solutions are far more prevalent than one might initially suspect.

Conservation of number is the *point de départ* for globally-convergent numerical homotopy methods. Most homotopy methods fit into a very general framework identified by Morgan and Sommese, called *coefficient parameter homotopy* [88], generalizing the so-called *cheater's homotopies* introduced by Li, Sauer, and Yorke [80]. The ethos of these parameter homotopies is very simple: *solve first a generic instance of Equation 1.1 (given by some $z_0 \in \mathbb{C}^m$), and then track solutions towards a specific instance (given by some $z_1 \in \mathbb{C}^m$) by deforming the generic problem into the specific problem.* The second step is accomplished by introducing a homotopy function

$$H(x,t) = f(x;\psi(t)), \tag{1.4}$$

where $\psi : [0,1] \to \mathbb{C}^m$ is a sufficiently regular path such that $\psi(0) = z_0, \psi(1) = z_1$, and such that there exists a *solution path* $x : [0,1) \to \mathbb{C}^n$ satisfying an initial value problem

$$\frac{d\,x}{d\,t} = -\left(\frac{d\,H}{d\,x}\right)^{-1} \frac{d\,H}{d\,t} \tag{1.5}$$

with the initial condition that $x(0)$ is some isolated, nonsingular solution to $f(x;z_0) = 0$. For a well-constrained problem and generic z_1, each of these solution paths converges to a unique solution $x(1)$ as $t \to 1$ such that $f(x(1);z_1) = 0$. In practice, approximate values of the solution path $x(t)$ along points t in some discretization of $[0,1]$ are obtained via numerical predictor/corrector methods. This is commonly called continuation, or *path-*

Algorithm 1 (**track**).
Input: z_0, z_1, Q_0: starting parameters, target parameters, and start solutions $Q_0 \subset X_{z_0}$
Output: target solutions $Q_1 \subset X_{z_1}$
for $x_* \in Q_0$ **do**
  Initialize: $t_* \leftarrow 0$
  while: $t_* < 1$ **do**
    update $t_* \leftarrow \min(t_* + \Delta t, 1)$
    predict $x_* \leftarrow x_* - (d_x H(x_*, t_*))^{-1} \; d_t H(x_*, t_*)$
    correct $x_* \leftarrow x_* - (d_x H(x_*, t_*))^{-1} \; H(x, t)$
  update $Q_1 \leftarrow Q_1 \cup \{x_*\}$
return Q_1

Figure 1.1: A simplified predictor/corrector path-tracking scheme.

tracking. For completeness, we give a very watered-down pseudocode for path-tracking the parameter homotopy of Equation 1.4, for a well-constrained system with as many unknowns as equations ($N = n$.) In practice, the step-size Δt is controlled adaptively so that $H(x_*, t_*) \approx 0$, multiple corrector steps and different predictor schemes are typically used, matrix inverses are replaced with linear-solves, and so-called endgame methods must be employed when the target solution set is singular or positive-dimensional or when solution curves diverge. See [5, 87, 103, 16] for an overview of numerical continuation methods and their application to polynomial system solving.

Parameter homotopies are appealing in the sense that the system specified by z_0 (aka the *start system*) has a similar structure to the system specified by z_1 (aka the *target system.*) But there is a chicken-egg problem—how do we solve the start system? In the context of parameter homotopies, this is referred to as the *offline* phase, since in principle it only needs to be done once per family of systems. For the offline phase, there are basically two alternatives:

- The first alternative is to track paths from some easily-solvable start system(s), which need not belong to the family given by Equation 1.1. The standard total-degree, multihomogeneous, polyhedral, and regeneration homotopies all fit into this paradigm. Despite their successes, these methods are frequently sub-optimal in the sense that they overestimate the number of solutions in X_{z_1}. The result is that extraneous solution

paths will diverge, an unnecessary computational overhead which also presents the numerical challenge of deciding when a solution path diverges. On a more philosophical note, the main limitation of these methods is that they do not fully exploit the structure of the family in Equation 1.1.

- In general, the systems $f(x; z_1) = 0$ and $f(x; z_0) = 0$ "look the same": we cannot expect solving one to be any easier than solving the other. Nevertheless, there are numerical homotopy methods that work directly with the family of interest which, under suitable assumptions, scale reasonably well with the cardinality of the generic fiber $d = \#X_{z_1}$. These methods are based on *monodromy*.

As a concept in mathematics, monodromy dates back to the 19th century, originating in foundational work of Riemann in complex analysis [94], and the Galois/monodromy groups of branched covers are implicit in the work of Jordan [65]. In more recent works from the field of symbolic-numeric computation, heuristics based on monodromy have enabled applications like computing Riemann matrices of algebraic curves [29] and polynomial factorization [44]. The latter application is closely related to the use of monodromy in numerical irreducible decomposition [101], in which points on algebraic varieties are classified according to irreducible components. A natural progression from the numerical irreducible decomposition was the use of monodromy as a heuristic for solving polynomial systems, appearing in several recent works in symbolc-numeric computation [33, 84, 57] and independently in the kinematic design literature [92, 13]. The closely-related problem of computing Galois/monodromy groups via numerical continuation was considered before in [79, 59]. Monodromy implementations are freely available in both the package `MonodromySolver` [32] for the computer algebra system Macaulay2 [46], which is used in this thesis, and the more recent `HomotopyContinuation.jl` [23], which achieves even greater efficiency through through the Julia language's native just-in-time compiler and a more sophisticated path-tracking algorithm [104].

1.2 Contributions of this thesis

In Chapter 2, we present some basic facts related to monodromy, and revisit a framework for solving polynomial systems using monodromy and homotopy continuation established in the author's previous publications [33, 20]. We describe some general considerations for efficiently using monodromy-based methods in polynomial system solving in the context of this framework. We illustrate some non-trivial aspects of this framework through several case studies using `MonodromySolver`, spanning subjects like computer vision, checmical reaction networks, Galois theory, kinematics, and numerical algebraic geometry. Each illustrates in a different way how monodromy can be a versatile tool for solving the polynomial systems that arise in a wide variety of applications.

Chapter 3 contains original results from the author's previous collaborations in computer vision [34, 40, 35]. The polynomial systems that arise are known in the literature as *minimal problems.* Monodromy allows us to compute the degree of each minimal problem, which serves as a proxy for the complexity of specialized *minimal solvers* which are used in RANSAC-based pipelines. We provide an overview of the classification of minimal problems for calibrated cameras with complete and missing data, and describe in additional detail how the monodromy computations were carried out. The results give another illustrative case study on the applications of monodromy, since a similar program of classifying minimal problems and computing their degrees can be carried over in principle to other settings (eg. non-calibrated cameras.)

In Chapter 4, we consider monodromy in settings where the problem of interest either decomposes into algebraic subproblems or possesses some kind of symmetry. The Galois/monodromy group, which can be heuristically computed using the framework of Chapter 2, tells us whether or not such a decomposition or symmetries exist. Decompositions and symmetries, once understood precisely, can be easily exploited for solving. Obtaining such a precise understanding is a difficult problem in general, which we solve for several of the problems occuring in Chapter 3. This chapter contains preliminary work which is joint with Viktor Korotynskiy, Tomas Pajdla, and Margaret Regan [36].

CHAPTER 2

MONODROMY

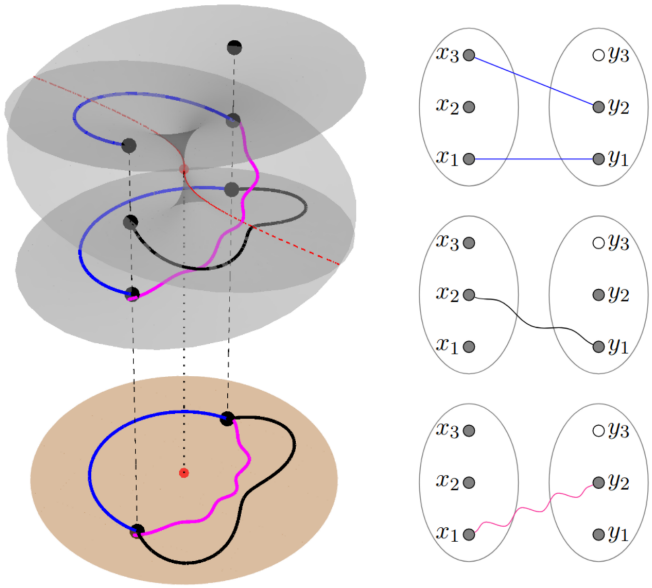


Figure 2.1: A graph of homotopies embedded in the base space of the branched cover of Example 1. Partial correspondences are drawn along each edge between the known solutions in grey. Taking x_3 as a seed solution, we subsequently discover x_2 and x_1 by numerical continuation along the appropriate paths in Z. Figure first appeared in [33].

The essence of monodromy as applied to polynomial system solving is the slogan *"collect all solutions starting from one."* This chapter brings some precision to this slogan. We give some mathematical background, and then revisit the graph-based framework first described in [33]. One of the main contributions of this framework was the simple observation that re-using information from monodromy loops is much more efficient than the so-called *naive dynamic strategy*, in which new loops are used in every iteration, allowing for monodromy-based blackbox solvers that are competetive with other leading homotopy methods. We further illustrate the efficiency of this framework by applying `MonodromySolver` to a variety of challenging examples.

2.1 Preliminary notions (Branched covers and Galois/monodromy groups)

Definition 2.1.1. A *branched cover* is a dominant, rational map $X \dashrightarrow Z$, where X and Z are irreducible algebraic varieties over $\mathbb{C}$ of the same dimension.

Readers who feel uncomfortable with the math should keep in mind that, in applications, $X \dashrightarrow Z$ is simply a map that is locally invertible at a generic data-point $z \in Z$, and defined by polynomial or rational functions. The varieties X and Z are said to be the *total space* and *base* of the branched cover, respectively. Depending on the context, we may also call Z the parameter space. The reader may safely assume that all varieties are quasiprojective.

Several consequences of Definition 2.1.1 deserve emphasis. Most importantly, *dominance* implies that for generic (and hence *almost all*) data $z \in Z$, the fiber over z, denoted X_z, is a nonempty, finite set. Second is the assumption of irreducibility. In principle, we can always reduce to the case of an irreducible variety by writing an arbitrary variety as the union of its irreducible components. On the other hand, in many applications, we are really interested in solutions which lie on an "interesting" irreducible component of some possibly reducible incidence variety. Finally, it is more natural to state various results using rational maps instead of regular maps. For instance, a branched cover might have a decomposition which is only valid on some Zariski-open subset of X, as is the case for Example 2. These decompositions are the subject of Chapter 4.

Pulling back rational functions from Z to X lets us identify $\mathbb{C}(Z)$ with a subfield of $\mathbb{C}(X)$. Since $\mathbb{C}(X)$ and $\mathbb{C}(Z)$ have the same transcendence degree over $\mathbb{C}$, the field extension $\mathbb{C}(X)/\mathbb{C}(Z)$ is finite. The *degree* of the map $X \dashrightarrow Z$ may be defined as the degree of this field extension. We write $\deg(X/Z)$ for this quantity, since the map $X \dashrightarrow Z$ is usually clear from context. We say that a nonempty Zariski-open $U \subset Z$ is a *regular locus* for $X \dashrightarrow Z$ if $U \cap Z_{\text{sing}} = \varnothing$ and if for all $z \in U$ the cardinality of the fiber X_z is equal to the degree of the map. The existence of such a U follows from basic results in algebraic geometry [99, cf. pp. 142].

We now recall the monodromy action on the fibers of of a degree-d branched cover $f : X \dashrightarrow Z$. Fix a regular locus U and a basepoint $z \in U$, and write $X_z = \{x_1, \ldots, x_d\}$. A

loop based at z is a continuous map $\gamma : [0,1] \to U$ which satisfies $\gamma(0) = \gamma(1) = z$. For each x_i, there exists a unique *lift* $\widetilde{\gamma}_i : [0,1] \to f^{-1}(U)$ satisfying $\gamma = f \circ \widetilde{\gamma}_i$ and $\gamma(0) = x_i$. This fact from topology is known as the *unique path-lifting property* (see eg. [54, Proposition 1.30]). The lifts based at each of the points $x_1, \ldots, x_d$ determine a permutation of the fiber, $\sigma_\gamma : X_z \to X_z$, which may be written in two-line notation as

$$\sigma_\gamma = \begin{pmatrix} x_1 & x_2 & x_3 & \cdots & x_d \\ \widetilde{\gamma}_1(1) & \widetilde{\gamma}_2(1) & \widetilde{\gamma}_3(1) & \cdots & \widetilde{\gamma}_d(1) \end{pmatrix}. \tag{2.1}$$

Remark 2.1.2. $Sym(X)$ denotes the symmetric group of all permutations from a finite set X to itself, S_n, A_n, denote symmetric and alternating groups acting on letters $[n] = \{1, \ldots, n\}$ (hence $Sym([n]) = S_n$), and C_n denotes a cyclic group of order n. At several points, it will be necessary to distinguish between abstract groups and the way they act on sets. For instance, the usual action of S_4 on $[4]$ is not equivalent to the action of $S_4 \hookrightarrow S_6$ on the $6 = \binom{4}{2}$ unordered pairs in $[4]$. The latter group may also described as $S_2 \wr S_3 \cap A_6$. Here, $\wr$ denotes the *wreath product*, whose basic properties are summarized in [96, Ch. 7]. For example, the permutation group $S_2 \wr S_3$ is isomorphic to the subgroup of S_6 consisting of all permutations that preserve the partition $[6] = [2] \cup \{3,4\} \cup \{5,6\}$.

One can show that the permutation σ_γ given in Equation 2.1 is independent of the homotopy class of γ in U, from which one obtains a homomorphism from the fundamental group $\pi_1(U, z)$ into the symmetric group $Sym(X_z)$:

$$\rho : \pi_1(U, z) \to Sym(X_z) \tag{2.2}$$

$$[\gamma] \to \sigma_\gamma. \tag{2.3}$$

The map ρ in Equation 2.2 is known as the *monodromy representation*, and its image is called the *monodromy group.* We will use either notation $\mathrm{Mon}(X/Z; U, z)$ or simply $\mathrm{Mon}(X/Z)$. The latter notation is justified by Proposition 2.1.4.

First, we recall the most basic and important fact about monodromy.

Proposition 2.1.3. The monodromy group acts transitively on the fiber X_z.

Proof. Suppose $x, x' \in X_z$. As an irreducible set in its *Zariski* topology, X is connected, and hence also path-connected, in its *complex* topology—see eg. [47, pp. 21–22]. Moreover, we maintain connectedness after excising any proper Zariski-closed $\Sigma \subset X$—that is, replacing $X \leftarrow X \backslash \Sigma$. This is because $\operatorname{codim}_{\mathbb{R}} \Sigma \geqslant 2$. If we take Σ containing all points where the branched cover is undefined or whose image is not contained in U, then there exists a path in $X \backslash \Sigma$ connecting x and x', which pushes forward to a monodromy loop γ such that $\sigma_\gamma(x) = x'$. □

It is well-known that monodromy groups are Galois groups. Proposition 2.1.4 makes this precise. We write $\mathbb{C}(X)^{\mathrm{gal}}/\mathbb{C}(Z)$ for the Galois closure of $\mathbb{C}(X)/\mathbb{C}(Z)$ and $\mathrm{Gal}(X/Z)$ for the Galois group of $\mathbb{C}(X)^{\mathrm{gal}}/\mathbb{C}(Z)$.

Proposition 2.1.4. Let $X \dashrightarrow Z$ be a branched cover with regular locus U, and fix a basepoint $z \in U$. Then the $\mathrm{Gal}(X/Z)$ and $\mathrm{Mon}(X/Z; U, z)$ are isomorphic as permutation groups. In particular, $\mathrm{Mon}(X/Z; U, z)$ is independent of the choice of (U, z).

A proof of Proposition 2.1.4 is given in [52]. In this proof, the Galois closure $\mathbb{C}(X)^{\mathrm{gal}}/\mathbb{C}(Z)$ is identified as an extension of $\mathbb{C}(X)$ obtained by adjoining certain germs of functions around points in X_z. With this identification, it is then argued (using the Galois correspondence and analytic continuation) that the Galois and monodromy actions on X_z coincide.

Example 1. Consider

$$X = \{(x,\, z) \in \mathbb{C} \times \mathbb{C} \mid x^3 = z\}$$

as a degree-3 branched cover over $Z = \mathbb{C}$ given by $(x,\, z) \mapsto z$. A regular locus is the punctured complex line $U = \{z \mid z \neq 0\}$. The monodromy group $\mathrm{Mon}(X/Z) \cong A_3 = C_3$ acts by cyclic permutation of $X_z = \{z, \omega\, z, \omega^2\, z\}$, where $\omega = \exp(2\pi i/3)$. Indeed, $\pi_1(U; z)$ is generated by the loop $\gamma(t) = e^{2\pi i t}$, which encircles the branch point $z = 0$ and induces the permutation σ_γ defined by

$$\sigma_\gamma = \begin{pmatrix} z & \omega\, z & \omega^2\, z \\ \omega\, z & \omega^2\, z & z \end{pmatrix}.$$

Example 2. Consider the "palindromic" family over $Z = \mathbb{C}^2$ given by

$$X = \{(x, a, b) \in \mathbb{C} \times \mathbb{C}^2 \mid x^4 + ax^3 + bx^2 + ax + 1 = 0\}.$$

The map $(x, a, b) \mapsto (a, b)$ is a branched cover of degree 4. A regular locus is given by $U = \mathbb{C}^3 \setminus V(\Delta)$, where $\Delta(a, b)$ is the discriminant

$$\Delta(a, b) = (2\,a - b - 2)\,(2\,a + b + 2)\left(a^2 - 4\,b + 8\right)^2.$$

Identifying the fundamental group $\pi_1(U)$ is less standard than in the previous example. This regular branched cover $X \to Z$ decomposes as a composition of rational branched covers $X \dashrightarrow Y, Y \dashrightarrow Z$, where

$$Y = \{(t, a, b) \mid a\,t^2 + b\,t + c - 2\,a = 0\}$$

is obtained by setting $t = (x^2 + 1)/x$. The monodromy action is equivalent to the action of the dihedral group D_8 on the vertices of a square, which is permutation isomorphic to the wreath product $S_2 \wr S_2 \hookrightarrow S_4$. In this example, $\mathbb{C}(X)/\mathbb{C}(Y)$ and $\mathbb{C}(Y)/\mathbb{C}(Z)$ are both Galois extensions, but $\mathbb{C}(X)/\mathbb{C}(Z)$ is not.

Example 3. Let

$$X = \{(x_1, x_2, x_3, a, b) \in \mathbb{C}^3 \times \mathbb{C}^2 \mid x^6 + a\,x^4 + b\,(x^2 + b)\}$$

$$Z = \mathbb{C}^2.$$

The map $(x, a, b) \mapsto (a, b)$ is a branched cover of degree 6, with a regular locus $U = \mathbb{C}^4 \setminus V(\Delta)$, where Δ is the discriminant

$$\Delta(a, b) = 64\,b^6\,\left(4\,a^3 - a^2 - 18\,a\,b + 27\,b^2 + 4\,b\right)^2.$$

We have $\mathrm{Mon}(X/Z) \cong S_2 \wr S_3 \cap A_6$. As *abstract groups*, there exists an isomorphism $S_2 \wr S_3 \cap A_6 \cong S_4$. The monodromy action is equivalent to the action of S_4 on 2-subsets

$\big\{\{i,j\} \mid 1 \leqslant i < j \leqslant 4\big\}$ which is induced by the standard action $S_4 \curvearrowright [4]$.

Example 4. Let

$$X = \{(x,a,b,c) \in \mathbb{C} \times \mathbb{C}^3 \mid x^6 + a\,x^4 + b\,x^2 + c\}$$

$$Z = \mathbb{C}^3.$$

As in the previous example, the map $(x,a,b) \mapsto (a,b)$ is a branched cover of degree 6. In this case we have that $\mathrm{Mon}(X/Z)$ is the full wreath product $S_2 \wr S_3$.

Example 5. Let

$$\widetilde{C} = \{(w,x) \in \mathbb{C}^2 \mid w^2 = (x^2-1)(x^2+1)\},$$

$$C = \{(y,z) \in \mathbb{C}^2 \mid y^2 = z(z-1)(z+1)\}.$$

The double cover $\widetilde{C} \to C$ given by $(w,x) \mapsto (wx, x^2)$ is *unramified*. Topologically, C is a punctured torus. The monodromy representation maps $\pi_1\left(C;(y,z)\right)$, a free group with two generators, onto $Sym\left(\widetilde{C}_{(y,z)}\right) \cong S_2$.

Definition 2.1.5. Two branched covers, $X_1 \dashrightarrow Z_1$ and $X_2 \dashrightarrow Z_2$, are *birationally equivalent* if there exist birational maps $X_1 \dashrightarrow X_2$ and $Z_1 \dashrightarrow Z_2$ such that the following diagram commutes:

$$\begin{array}{ccc} X_1 & \dashrightarrow & X_2 \\ \vdots & & \vdots \\ \downarrow & & \downarrow \\ Z_1 & \dashrightarrow & Z_2. \end{array}$$

Proposition 2.1.6. The Galois/monodromy group of a branched cover is a birational invariant.

Proof. This follows easily from Proposition 2.1.4, since a birational equivalence of branched covers induces an isomorphism of field extensions. A more topological proof is also possible. Let us write $\Psi : X_1 \dashrightarrow X_2$, $\psi : Z_1 \dashrightarrow Z_2$ for the horizontal maps appearing in Definition 2.1.5. Then, for suitable regular loci $U_1 \subset Z_1, U_2 \subset Z_2$, there is an isomorphism $\mathrm{Mon}(X_1/Z;U_1,z) \cong \mathrm{Mon}(X_2/Z;U_2,\Psi(z))$ defined by identifying, for $\gamma : [0,1] \to U$ based at z, the lifts $\widetilde{\gamma}_1, \ldots, \widetilde{\gamma}_d$ in X_1 with the lifts $\Psi \circ \widetilde{\gamma}_1, \ldots, \Psi \circ \widetilde{\gamma}_d$ in X_2. □

2.2 A graph-based framework for monodromy computations

Proposition 2.1.3 suggests that, when our family of polynomial systems is modeled by a branched cover, it is possible to recover all solutions to a generic problem instance starting from just one solution. To do this, it suffices to conjure up loops $\gamma_1, \ldots, \gamma_k$ such that the subgroup $\langle \sigma_{\gamma_1}, \ldots, \sigma_{\gamma_k} \rangle \leqslant \mathrm{Mon}(X/Z)$ acts transitively. However, in practice we have a very limited knowledge of the main mathematical objects in play—the regular locus $U \subset Z$, the fundamental group $\pi_1(U; z)$, perhaps even the irreducible varieties X and Z. In light of this lack of knowledge, it benefits us to make very minimal assumptions about how to give our branched cover $X \dashrightarrow Z$ as input to a procedure such as Algorithm 2.

In the remainder of this chapter, we primarily consider branched covers of affine varieties $X \to \mathbb{C}^m$ given by coordinate projection $(x, z) \mapsto z$, where

$$\mathcal{I}_X = \langle g_1(x; z), \ldots, g_k(x; z) \rangle \subset \mathbb{C}[x_1, \ldots, x_n, z_1, \ldots, z_m] \tag{2.4}$$

is a prime ideal in both variables x and parameters z. The main examples considered in this thesis are of this form, at least up to birational equivalence. These problems are well-posed in the sense that exact solutions exist for generic parameter values. We remark that for a branched cover $X \dashrightarrow Y$ where $Y \subset \mathbb{C}^m$ is a proper subvariety (for instance, as in Example 5), we could take a generic projection $Y \dashrightarrow Z$ onto an affine space $Z = \mathbb{C}^{\dim Y}$, and use Proposition 4.1.2 to identify $\mathrm{Mon}(X/Y)$. Using this approach to study over-constrained problems is an intriguing possibility, but beyond the scope of this thesis.

Since our knowledge of the irreducible variety X may be limited, it is very useful to observe that we do not need to know all generators of $\mathcal{I}_X$. To "do monodromy", we only really need to know two things:

1) the ability to sample a generic point $(x^*, z^*) \in X$, and

2) a set of n equations vanishing on X,

$$f(x;z) = f(x_1,\ldots,x_n;z_1,\ldots,z_m) = \begin{bmatrix} f_1(x_1,\ldots,x_n;z_1,\ldots,z_m) \\ \vdots \\ f_n(x_1,\ldots,x_n;z_1,\ldots,z_m) \end{bmatrix}, \quad (2.5)$$

such that the Jacobian $d_x f(x^*;z^*)$ is an invertible $n \times n$ matrix. We say that equations Equation 2.5 form a *well-constrained system* for the branched cover $X \to \mathbb{C}^m$.

The point (x^*, z^*) is called the *seed pair* for Algorithm 2. The well-constrained system f may consist of polynomial or rational functions. For generic $(x^*, z^*) \in X$ and generic $z \in \mathbb{C}^m$, the segment $\psi_1 : [0,1] \to \mathbb{C}^m$ defined by the straight-line

$$\psi_1(t) = (1-t) \cdot z^* + t \cdot z \quad (2.6)$$

will have a lift $\widetilde{\psi_1}(t)$ to X based at $\widetilde{\psi_1}(0) = (x^*, z^*)$. An approximation of the lifted path can be computed by numerical path-tracking.

Proposition 2.2.1. For generic z_*, z, the lifted path $\widetilde{\psi_1} : [0,1] \to X$ will not intersect the exceptional subvariety of X where $d_x\, f$ is singular or f is undefined for any $t \in [0,1]$.

Proof. This is the same argument as Lemma 7.1.2 in [103]. Let Σ be the image of the exceptional set under projection onto $\mathbb{C}^m$. We have $\dim_{\mathbb{C}} \Sigma \leqslant m-1$, and hence also $\dim_{\mathbb{R}} \Sigma \leqslant 2m-2$. We may assume that neither z nor z^* are in Σ. Consider the exceptional set which is the union of all *real lines* (ie., lines parametrized as in Equation 2.6) which contain z^* and a point in Σ. This set has real dimension at most $2m-1$, so it suffices to assume that z lies outside of this exceptional set. □

Thus, if we consider the *parameter homotopy*

$$H(x,t) = f\left(x;\psi_1(t)\right) = 0, \quad (2.7)$$

Proposition 2.2.1 implies that the solution curves $x(t)$ will satisfy $H(x(t),t) = 0$ and the initial condition $x(0) = x^*$ will stay on our irreducible variety X with probability-one, and

hence $\widetilde{\psi_1}(t) = (x(t), \psi_1(t))$. Thus, although the variety X need not be a complete intersection, appropriate use of a well-constrained system enables us to compute solutions on X using the same number of equations as unknowns. In practice, our numerical approximations to $\widetilde{\psi_1}(t)$ remain "close" to X with some probability that depends on the conditioning and the implementation of the numerical methods. Our previous remarks still apply if we replace the straight-line segment $\psi_1(t)$ by some other suitably generic path in Z. By numerically continuing solutions along some path $\psi_2(t)$ from z to z^*, and then along some other path $\psi_1(t)$ from z^* to z, the concatenated path $\gamma = \psi_2 * \psi_1$ is a loop based at z which induces a monodromy permutation σ_γ.

To organize the discovery of new solutions, it is natural to collect all homotopies used together in a finite, connected, undirected graph G, in which every vertex in $V(G)$ is a point $z \in \mathbb{C}^m$, and every edge $e = (z_0, z_1) \in E(G)$ is decorated with a homotopy. To specify the homotopies, we orient each edge $\vec{e} = \overrightarrow{z_0 z_1}$ and define

$$H_{\vec{e}}(x; t) = f(x; \psi_{\vec{e}}(t)) \tag{2.8}$$

where $\psi_{\vec{e}} : [0, 1] \to \mathbb{C}^m$ is some sufficiently regular path such that $\psi_{\vec{e}}(0) = z_0, \psi_{\vec{e}}(1) = z_1$. For the reverse orientation, set $H_{\overleftarrow{e}}(x; t) = H_{\overleftarrow{e}}(x; 1 - t)$. Thus, any vertex in the graph may serve as a start system or a target system, depending on the state of the computation.

Depending on how the equations f depend on the parameters z, we may flexible on how the path ψ is represented. For instance, if f is linear in parameters in the sense that

$$f(x; z_1 + \psi_1 z_2) = f(x; z_1) + \alpha_1 f(x; z_2),$$

the straight-line homotopies proposed in [33] are given by $\psi = \psi_1 * \psi_2 * \psi_3$, where

$$\begin{aligned}
\psi_1(t) &= \exp(i\,t)\, z_0, && t \in [0, R_0] \\
\psi_2(t) &= \exp(i\,R_0)\, z_0 + \exp(i\,R_1)\, z_1, && t \in [0, 1] \\
\psi_3(t) &= \exp(i\,t)\, z_1, && t \in [R_1, 0],
\end{aligned}$$

where $R_0, R_1 \sim \mathrm{Unif}([0, 2\pi])$. This is the simplest instance of an *endpoint randomization*

scheme. A cartoon illustration is provided in Figure 2.2. Two edges in the homotopy graph are drawn on the left, and the corresponding monodromy loop is drawn on the right. A fictional discriminant locus is drawn in red.

The endpoint randomization scheme discussed above is a special case of the well-known "gamma-trick" (cf. [103, Ch. 7].) It is possible to devise analagous randomization schemes for other types of families. For instance, in Chapter 3 we will encounter polynomial systems which are equivariant with respect to a group action $\mathcal{G} \curvearrowright \mathbb{C}^m$ in the sense that

$$f(x; z) = f(x; g\, z) \qquad \forall g \in \mathcal{G}.$$

The "gamma-trick" above corresponds to the case where $\mathcal{G}$ is the group of complex numbers with modulus 1, acting by coordinate-wise multiplication. Intuitively—the bigger the group $\mathcal{G}$, the more efficiently we can explore the parameter space $\mathbb{C}^m$. Of course, the main difficulty in devising such a scheme is that it requires us to know something extra about the structure of our family. Nevertheless, endpoint randomization allows us to connect the fibers X_{z_1} and X_{z_2} with different correspondences using the same basic type of edge homotopy, such as the linear segment in Equation 2.6. Thus, endpoint randomization allows us to introduce multiple edges between distinct vertices without computing redundant correspondences between the fibers over z_0 and z_1. Our graph G can actually be a multigraph, depending on how the homotopies in Equation 2.8 are constructed. Using multiple edges has the potential to be more memory-efficient, since it is somewhat less expensive to store correspondences between solutions than the solutions themselves.

Ultimately, storing correspondences between solutions is the main feature that distinguishes our graph-based monodromy framework from approaches previously considered in the literature. To describe the basic algorithm, it is convenient to define the state of a partially known homotopy graph. The state of a homotopy graph G is defined by the known solutions Q_z for each vertex $z \in V(G)$, and the correspondence set C_e for each edge $e = (z_0, z_1) \in E(G)$. Upon termination of Algorithm 2, each correspondence set C_e is a bijective matching between known solutions in Q_{z_0} and Q_{z_1}. Thus, any closed walk starting at $z \in V(G)$ determines a permutation $\sigma_\gamma : Q_z \to Q_z$, where the loop γ is obtained by

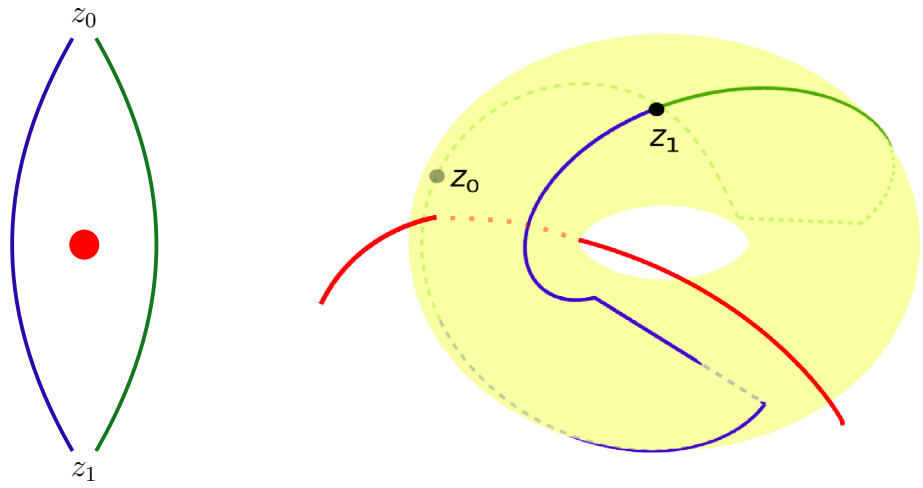


Figure 2.2: Endpoint randomization via the γ-trick.

concatenating the paths $\psi_{\vec{e}}$ defining each edge homotopy in Equation 2.8.

Having defined the state of a homotopy graph, we may finally state the basic monodromy algorithm. Upon termination of Algorithm 2, the graph G returned by encodes a group $M_G \curvearrowright \{X_z\}$ generated by loops in the edge-wide embedding of G in $Z = \mathbb{C}^m$ given by $\vec{e} \mapsto \psi_{\vec{e}}([0,1])$. These loops are easily computed from a spanning tree of G. Assuming the embedded graph avoids the exceptional set Σ of Proposition 2.2.1 and no numerical failures occur in the subroutine **track**, we give an easily-verified correctness statement for Algorithm 2 in the form of Proposition 2.2.2.

Proposition 2.2.2. When Algorithm 2 terminates, the set Q_1 contains the the orbit of (x^*, z^*) under M_G. Generators of the group M_G can be computed by composing bijective correspondences C_e along edges in closed walks in G obtained from joining z^* to a set of basic cycles for G.

2.3 Uncertainty in monodromy computations

2.3.1 Probabilistic models and complexity

To harness the power of monodromy for solving nontrivial examples of polynomial systems, it seems the best we can hope for is some sort of probabilistic technique. Likewise for the problem of computing the monodromy group itself—although a more deterministic

Algorithm 2 (Monodromy meta-algorithm)**.**
Input: $(G, (x_*, z_*))$

G — a graph with vertices $z_1, \ldots, z_{n_G} \in Z = \mathbb{C}^m$ and edges decorated by homotopies $H_{\vec{e}}$ obtained from a well-constrained system for $X \to Z$,

$(x_*, z_*) \in X$ is an initial seed such that $z_1 = z_*$.

Output: Solutions in X_z for all $z \in V(G)$ and correspondences C_e for all $e \in E(G)$
Initialize: $Q_1 = \{x_*\}$, $Q_2 = \cdots = Q_{n_G} = \emptyset$, $C_e = \emptyset \ \ \forall e \in E(G)$
while $\exists\, e = (z_i, z_j) \in E(G)$ with $\#C_e < \max(\#Q_i, \#Q_j)$ **do**
  select a directed edge $\overrightarrow{e} = \overrightarrow{z_i\, z_j}$ from $(z_i, z_j) \in E(G)$ with $\#Q_i > \#C_e$
  track points in Q_i to points in Q_j along $H_{\vec{e}}$
  update C_e and Q_j with undiscovered correspondences and solutions
return $\left(\{Q_v\}_{v \in V(G)},\ \{C_e\}_{e \in E(G)}\right)$

Figure 2.3: Meta-algorithm for monodromy of a branched cover $X \to Z$.

approach is given by the branch-point method of [59], this method unfortunately requires computing the intersection of some discriminant locus with a generic line in $\mathbb{C}^m$. This is typically costly when compared to Algorithm 2.

In light of Proposition 2.2.2, if we knew the root count $d = \deg(X/Z)$ in advance, we could run Algorithm 2 until it is reached, augmenting the graph G within the **while** loop with new edges and vertices as necessary. However, in many cases we do not know the root count *a priori.* Thus, it is essential to consider the following question: how many loops are needed to generate a transitive action? In terms of the graph G—how large should its first Betti number $\beta_1(G)$ be? The answer to this question clearly depends on the Galois/monodromy group. For instance, if we consider the "corner-case"

$$x_i^2 = z_i,\ i = 1, \ldots n,$$

then the Galois/monodromy group $(C_2)^n$ requires n generators at a minimum, and no smaller number will generate a transitive subgroup. On the other hand, we frequently have that the Galois/monodromy group is full-symmetric—$\mathrm{Mon}(X/Z) \cong S_d$. In this case, it is actually quite reasonable in practice to take a small number of loops—in other words, to have $\beta_1(G) = O(1)$ as $d \to \infty$. We observe this experimentally in section 2.4 on

several problems of interest. Here, we validate this observation with an extremely simple probabilistic model, which assumes that the group M_G is generated by j permutations $\sigma_1, \ldots, \sigma_j$ drawn *independently and uniformly at random* from S_d. In other words, we consider $\sigma_1, \ldots, \sigma_j \sim \mathrm{Unif}(S_d)$ as i.i.d. random variables. The accuracy of this model primarily depends on the underlying scheme for generating random loops $\gamma_1, \ldots, \gamma_j$ as well as the nontrivial assumption $\mathrm{Mon}(X/Z) \cong S_d$.

The uniform model is highly over-simplied. However, it already suggests a major difference between Algorithm 2 and a *naive, dynamic strategy* that discards the knowledge obtained from previous loops. As observed in [33, 92], such a strategy is naturally modeled by the *coupon collector's problem*, requiring $d \log d$ trials in expectation. Maintaining a set of partial permutations yields a modest asymptotic improvement: provided that the monodromy group is full-symmetric, the expected number of path-tracks under the uniform model is $O(d)$. The following result appearing in [33] is a generalization of a celebrated result by Dixon [31, Theorem 2], who considered the case $j = 2$. In our paper, we give a proof which follows quite closely the same proof Dixon gave when $j = 2$.

Theorem 2.3.1. For $j \geqslant 2$, we have

$$\Pr\left[\sigma_1, \ldots, \sigma_j\rangle \text{ is transitive for } \sigma_1, \ldots, \sigma_j \sim \mathrm{Unif}(S_d)\right] = 1 - d^{1-j} + O\left(d^{-j}\right).$$

We refer to [33] for a complete proof and further discussion of this asymptotic result. To complement the asymptotics, Table 2.1 contains exact values of the success probabilities appearing in Theorem 2.3.1 for various values of d and j. Under the uniform model, $j \approx 2$ loops are sufficient to generate a transitive action, in expectation. To derive the table, let t_d denote the probability appearing in Theorem 2.3.1. Suppose we partition the set of letters $[d]$ into k_i parts of size i for each $1 \leqslant i \leqslant d$. Letting

$$K_d = \left\{\vec{k} = (k_1, \ldots, k_d) \in \mathbb{N}^d \mid \sum i\, k_i = d\right\},$$

the number of partitions corresponding to each $\vec{k} \in K_d$ is $d!/(\prod_{i=1}^{d}(i!)^{k_i} \cdot k_i!)$.

For each $\vec{k} \in K_d$, the partition of $[d]$ into the orbits of the group $\langle\sigma_1, \ldots, \sigma_j\rangle$ is $\vec{k}$-indexed

precisely when this group acts transitively on all classes of some partition associated to $\vec{k}$. The number of tuples in S_i^j with coordinates generating a group acting transitively on $\{1, \ldots, i\}$ is $t_i\,(i!)^j$. Thus, we may count the j-fold product $S_d \times \cdots \times S_d$ as

$$\begin{aligned}(d!)^j &= \sum_{\vec{k} \in K_d} \frac{d!}{\prod_{i=1}^d (i!)^{k_i} \cdot k_i!} \cdot \prod_{i=1}^d \left(t_i\,(i!)^j\right)^{k_i} \\ &= d! \cdot \sum_{\vec{k} \in K_d} \prod_{i=1}^d \frac{\left(t_i\,(i!)^{j-1}\right)^{k_i}}{k_i!}.\end{aligned}$$

Let $\widehat{F}(x)$ denote the generating function the sequence $\left((d!)^{j-1}\right)_{d=1}^{\infty}$. Our counting formula implies the formal identity

$$\begin{aligned}\sum_{d=1}^{\infty} d \cdot (d!)^{j-1}\, x^{d-1} &= \frac{d}{dx}\,\widehat{F}(x) \\ &= \frac{d}{dx}\,\exp\left(\sum_{i=1}^{\infty} t_i\,(i!)^{j-1} x^i\right) \\ &= \left(\sum_{d=0}^{\infty} (d!)^{j-1}\, x^d\right) \cdot \sum_{i=1}^{\infty} i \cdot t_i\,(i!)^{j-1} x^{i-1} \\ &= \sum_{d=1}^{\infty} x^{d-1} \left(\sum_{i=1}^{d} i \cdot t_i \left(i! \cdot (d-i)!\right)^{j-1}\right)\end{aligned}$$

Equating coefficients of x^{d-1} yields

$$d = \sum_{i=1}^{d} \binom{d}{i}^{1-j} i\, t_i,$$

giving linear equations which can be solved successively for $t_1,\, t_2,\, \ldots$.

The uniform permutation model supports the following hypothesis—for a branched cover with Galois/monodromy group S_d, the probability that Algorithm 2 yields all solutions in the fiber X_{z*} approaches 1 very rapidly as the Betti number of the underlying graph grows slightly. We do not comment on the extent to which the bounds of Theorem 2.3.1 are sharp. Nor do we treat the analagous question of when Algorithm 2 terminates with the correct Galois/monodromy group, although precise statements are known when $j = 2$ due

Table 2.1: The probability that j random permutations generate a transitive subgroup of S_d.

d	$j = 2$	$j = 3$	$j = 4$
1	1	1	1
2	0.75	0.875	0.9375
3	0.72222222	0.89814815	0.96450617
4	0.73958333	0.93012153	0.98262080
5	0.76833333	0.95334722	0.99115752
10	0.88180398	0.98954768	0.99898972
20	0.94674288	0.99747856	0.99987487
30	0.96536852	0.99888488	0.99996295

to Babai [10] and subsequent simplifications by Eberhard and Virchow [38].[1]

Looking beyond the uniform model and the restrictive assumption that $\mathrm{Mon}(X/Z) \cong S_d$, we may consider an arbitrary scheme for generating loops which induces a probability distribution $P : \mathrm{Mon}(X/Z) \to [0, 1]$. The random permutations $\sigma_k \in \mathrm{Mon}(X/Z)$ obtained by concatenating k independent random loops form a Markov chain with probability distribution given by the k-fold convolution $P^{*k} : \mathrm{Mon}(X/Z) \to [0, 1]$. Under the very mild assumption that P is not supported on a proper coset of $\mathrm{Mon}(X/Z)$, the distribution P^{k*} converges in total variation to the uniform distribution on $\mathrm{Mon}(X/Z)$ (see eg. [30, Ch. 4, Theorem 3].) This suggests the intriguing possibility of using Theorem 2.3.1 to give rigorous, probabilistic guarantees for some variant of Algorithm 2, provided that one could devise an explicit process for generating loops $\gamma_1, \ldots, \gamma_k$ that allows one to bound the rate of convergence of P^{k*}. Even more ambitiously, one could demand that such a procedure be efficient in a complexity-theoretic sense: for instance, by requiring it to run in time $\mathrm{poly}(d)$ (with other parameters fixed) with bounded probability of failure. This raises the further issue of bounding the expected runtime of **track** and rigorously certifying its correctness. Prior work on Smale's 17th problem [18, 25, 73] and certified path-tracking [17, 55, 108] should convince us that this is no easy feat. Still, the author believes that rigorous explanation of monodromy's successes deserves attention from people with the right tools to attack this difficult and open-ended problem.

[1] Babai's proof uses the classification of finite simple groups.

2.3.2 Dealing with failures

In reality, the subroutine **track** in Algorithm 2 must operate with floating-point arithmetic. Although the edges in the graph will avoid the discriminant locus with probability-one, they will get close to it. Thus, **track** will fail with some positive probability, which may be rather high for poorly conditioned systems. In principle, the failure probability can be driven to 0 by working with more than the 53 bits (≈ 16 digits) provided by the standard IEEE `double`. However, this comes at a disproportionate cost, since higher-precision floating point is typically not implemented at the hardware level.

An appealing aspect of the graph-based framework is that it is robust to a moderate number of failures. If the goal is simply to collect all solutions starting from one, then we might tolerate a few failures along each edge, so long as some vertex eventually accumulates $\deg(X/Z)$-many solutions. This should be contrasted with a single start-system approach, in which any failing path must be adaptively re-run with more conservative tolerances until it succeeds. Running `monodromySolve` with the option `Verbose=>true` will print a number of messages to the screen when failures occur:

> **tracking failure:** these failures are classified according to the implementation of `track` and `trackHomotopy` in Macaulay2, including cases where the minimum step-size is reached, or when a solution curve gets too close to infinity or being singular.
>
> **correspondence conflict:** tracking a solution in Q_i along $\vec{e}$ to Q_j results in a solution already occuring in the correspondence C_e. This suggests the unfortunate possibility of path-jumping, where the path-tracker unknowingly jumps between different solution curves. Another possibility is that the problem is poorly-conditioned, and our numerical approximations cannot be distinguished under the chosen tolerances.
>
> **filter failure:** using the option `filterCondition`, the output of **track** can be post-processed according to some filter function. This can be used to guard against path-jumping—when the well-constrained system has multiple irreducible components besides X, numerical approximations to the lifted paths have the potential to wrongly

discover solutions on these excess components, effectively contaminating the entire computation. Sometimes we know enough to test when newly discovered solutions lie on these excess components; when they do, we do not store them in Q_j.

2.3.3 When to stop?

Depending on our goal, different stopping criteria are available for monodromy. The stopping criterion of Algorithm 2 ensures that we can compute permutation generators for the group M_G. However, if d is known *a priori* and our only goal is to find d solutions, we can simply terminate the computation once the target solution count is attained at some vertex. This type of stopping criteria can also be applied to showing that the degree of a problem is lower-bounded by some integer, as we do later in section 3.3.

When the root count is not known *a priori*, the exhaustive stopping criterion of Algorithm 2 is the most straight-forward option. In practice, we also often limit the number of iterations with no new solutions discovered—the default limit in `MonodromySolver` is 10. A third option is based on *numerical trace tests* [102, 58, 78], which give effective criteria for the completeness of solution sets. The various trace tests all amount to checking that a suitably-defined function called the trace is linear. In the original setting of [102], the trace test could be applied to a branched cover of the form $X \to \mathbb{G}_{k,n}$, where $X \subset \mathbb{P}^n \times \mathbb{G}_{k,n}$ is an incidence correspondence of points on a codimension-k subvariety $X' \subset \mathbb{P}^n$ and the k-planes in the Grassmannian $\mathbb{G}_{k,n}$ that contain them. The degree of this branched cover is the degree of the projective variety $\deg X'$, and the monodromy group of such a branched cover is full-symmetric (see eg. [6, pp. 111–112].) This trace test can be reduced to the case where X' is a curve ($k = 1$.) It amounts to taking parallel slices H_1, H_2, H_3 in some chart on $\mathbb{P}^n$ and checking that the known points in $H_i \cap X'$ move linearly—this occurs iff the number of known points equals $\deg(X')$. Checking linearity of the trace reduces to numerical linear algebra. In practice, this computation is not rigorously certified; even if we know the points to arbitrary precision through Newton refinement, we cannot easily say that the trace function is *exactly* linear.

For more general branched covers, the multihomogeneous trace tests developed in [58, 78] must be applied. For a branched cover $X \to \mathbb{P}^m$ with $X \subset \mathbb{P}^n \times \mathbb{P}^m$, the complex-

ity of these trace tests depend on the multi-degrees $d_{(m,0)}(X), \ldots, d_{(0,m)}(X)$. Here, the degree of $X \to \mathbb{P}^m$ equals the multi-degree $d_{(0,m)}(X)$. The trace test of [58] requires $O\left(\sum_{k=0}^{m} \binom{m}{k} d_{(k,m-k)}(X)\right)$ path-tracks. The trace test of [78] requires a more modest $O\left(d_{(0,m)}(X) + d_{(1,m-1)}(X)\right)$ path-tracks. Still, the multidegrees form a log-concave sequence by Hodge-theoretic results due Khovanskii and Teissier (see [77, Ch 1.6] and the references therein), and we can reasonably expect $d_{(1,m-1)}(X)$ to be significantly larger than $d_{(0,m)}(X)$. Therefore, on problems of larger degree, the trace test should be used prudently.

2.4 Case Studies

2.4.1 Monodromy as a blackbox solver

A system with as many equations as unknowns is usually said to be *square*. We expect such a system to have finitely many solutions. Several theorems bound the number of isolated solutions of a square system. The most classical is Bézout's theorem, where the bound is the product of the polynomials' degrees. A sharper result in many cases is the celebrated Bernstein-Khovanskii-Kushnirenko (BKK) theorem [70, 19], where the bound is given by the mixed volume of the polynomials' Newton polytopes. Example 6 and the examples of Table 2.2 and subsection 2.4.3 illustrate that there may still be a large discrepency between these bounds and the actual number of solutions.

Table 2.2: Wall time in seconds of `MonodromySolver` vs other solvers.

problem	wnt	$SO(4)$	$SO(5)$	$SO(6)$	$SO(7)$
# vars	19	16	25	36	49
BKK bound	60	1024	32768	2097152	268435456
degree	9	40	384	4768	111616
MonodromySolver	0.52	4	23	528	42791
Bertini	42	81	10605	out of memory	
PHCpack	862	103	$>$ one day		

Table 2.2, taken from [33], gives a comparison of `MonodromySolver` against two other leading polynomial homotopy continuation solvers, PHCPack and Bertini [106, 15].

Related experiments may be found in [33, 20]. PHCpack uses the polyhedral homotopy [63], which tracks a number of paths equal to the BKK bound. The Bertini runs were using an equation-by-equation method known as regeneration [62], where the number of tracked paths is harder to predict. On these examples, `MonodromySolver` significantly outperformed PHCpack and Bertini, by virtue of tracking fewer paths and avoiding a potentially expensive mixed volume computation. We note that although Bertini dominates PHCpack on these examples, the situation can be very much reversed for other problems. We also note that the implementations of polyhedral homotopy and monodromy in `HomotopyContinuation.jl` [23] likely outperform their counterparts in Table 2.2 in the regime where the degree is large and compilation overhead is negligible. The column labeled "wnt" refers to a suitably randomized system of equillibria for a chemical reaction network that is studied in cell-signaling—see [33, 48] for more details. The columns labeled $\mathrm{SO}(n)$ refer to computing the degree of the special orthogonal group. A well-constrained system is obtained from $n(n+1)/2$ upper-diagonal entries of $\mathbf{R}\mathbf{R}^\top - \mathbf{I} = 0$ together with $n(n-1)/2$ inhomogeneous linear equations. In [21], `MonodromySolver` was used to check a closed form for the degree obtained from general results of Kazarnovskii [66]. In addition to the research presented in this thesis, we note the use of `MonodromySolver` in applications as diverse as graph rigidity [12] and quantum physics [107].

It can be difficult to get a fair comparison between solvers. This difficulty is compounded when we are faced with a well-constrained problem with more equations than unknowns. Blackbox solvers will regularize such a system by taking random linear combinations of the equations (called randomization or "squaring up"), adding random linear combinations of new slack variables, or simply dropping equations. Any of these strategies has the undesirable effect of introducing extraneous solutions. Monodromy, when initialized from a generic seed pair on the component of interest, will avoid these extraneous solutions with probability 1. Throughout this thesis, we reduce systems with more equations than unknowns to a well-constrained system with Algorithm 3. This is just a numerically-sensible version of the usual greedy algorithm. Orthogonal projection is done via SVD—this is not a bottleneck, since generally $n < 100$.

Example 6. We consider a problem from computer vision—reconstruction of nine lines

Algorithm 3 (square-down).
Input: seed pair (x_*, z_*), system $f = (f_1(x; z), \ldots, f_N(x; z))$
Output: square system $\hat{f} = (f_{i_1}, \ldots, f_{i_n})$ with $\operatorname{rank} d_x \hat{f} = n$
  Initialize: $\mathbf{v}_1, \ldots, \mathbf{v}_N \leftarrow$ rows of $d_x \hat{f}$, $\mathcal{B} \leftarrow \varnothing$, $i \leftarrow 1$
  while: $\#\mathcal{B} < n$ **do**
    $\mathbf{v}_i \leftarrow \mathbf{v}_i - \operatorname{proj}_{\mathcal{B}}(\mathbf{v}_i)$
    if: $\|\mathbf{v}_i\|^2 >> 0$ **then** $\mathcal{B} \leftarrow \mathcal{B} \cup \{\mathbf{v}_i\}$
    $i \leftarrow i + 1$
return $(f_{i_1}, \ldots, f_{i_n})$ w/ $\mathcal{B} = \{\mathbf{v}_{i_1}, \ldots, \mathbf{v}_{i_n}\}$.

Figure 2.4: Computing a well-constrained subsystem from the seed pair.

from three uncalibrated views. We use the formulation of Larsson et al. [75], where two of the lines have been fixed up to projective change of coordinates. In this formulation, there are 14 unknown entries in three camera matrices:

$$P_1 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \tag{2.9}$$

$$P_2 = \begin{pmatrix} x_1 & 1 & 0 & -1 \\ 0 & x_2 & x_3 & -x_3 \\ x_4 & x_5 & x_6 & x_7 \end{pmatrix} \tag{2.10}$$

$$P_3 = \begin{pmatrix} x_8 & x_9 & 0 & -x_9 \\ 0 & x_{10} & 1 & -1 \\ x_{11} & x_{12} & x_{13} & x_{14} \end{pmatrix}. \tag{2.11}$$

The 63 parameters are the homogeneous equations of seven lines in three views:

$$\mathbf{l}_{1,1}, \mathbf{l}_{1,2}, \mathbf{l}_{1,2}, \ldots, \mathbf{l}_{7,1}, \mathbf{l}_{7,2}, \mathbf{l}_{7,3} \in \mathbb{C}^{3\times 1},$$

These lines correspond in the sense that certain 4×3 matrices drop rank:

$$\operatorname{rank} \left(\begin{array}{c|c|c} P_1^\top \mathbf{l}_{i,1} & P_2^\top \mathbf{l}_{i,2} & P_3^\top \mathbf{l}_{i,3} \end{array} \right) \leqslant 2, \qquad i = 1, \ldots, 7 \tag{2.12}$$

The maximal minors of matrices in Equation 2.12 give us polynomial equations

$$f_1(\mathbf{x};\mathbf{l}) = \ldots = f_{28}(\mathbf{x};\mathbf{l}) = 0. \tag{2.13}$$

For generic parameter values, these polynomials have 36 solutions and 4 distinct Newton polytopes. To get a well-constrained system $f_{i_1}, \ldots, f_{i_{14}}$ for the branched cover $V(f_1, \ldots, f_{28}) \to \mathbb{C}^{63}$, we could take two minors corresponding to each of the given rank constraints. Bézout's theorem predicts $2^{14} = 16,384$ solutions for these subsystems, whereas their mixed volumes depend on the selection of minors. For generic parameters, permuting such a selection among the 7 rank constraints does not change the mixed volume, so representative mixed volumes for the square subsystems may be obtained by placing 7 unlabeled balls into $6 = \binom{4}{2}$ labeled bins. In doing so, we obtain 792 square subsystems, whose mixed volumes, as computed by Gfan [64], range from 470 to $2,858$. Another option is to *randomize* the system by taking 14 generic linear combinations of $f_1, \ldots, f_{28}$. In this case, the mixed volume is simply the normalized volume of the Newton polytope of such a linear combination, which turns out to be $3,328$.

For this example, we performed a short computational experiment to illustrate the behavior of `MonodromySolver` when working with more equations than unknowns. Timings in Table 2.3 are given in seconds. We used Algorithm 3 to select a square subsystem. To generate the seed pair, we choose values for $x_1, \ldots, x_{14}$ at random from the complex unit circle and taking each $\mathbf{l}_{i,j}$ to span the kernel of $(P_i\,\mathbf{v}_{j,1}|P_i\,\mathbf{v}_{j,2})^\top$, where $\mathbf{v}_{j,1},\ \mathbf{v}_{j,2}$ are points on a fabricated "world line." In all cases, a graph with two vertices and four edges was used, and the computation terminates with the same number of solutions at both nodes. However, with the default path-tracker settings, this number was occassionally higher than the true degree 36 due to path-jumping. To make the monodromy more robust, we can either use more conservative tracker settings—here `tStepMin => 1e-8`, `maxCorrSteps => 2`—or use a filter function—here, we do not store any newly discovered solution such that the singular values of any of the 7 rank-constrained matrices exceeds 10^{-5}.

Table 2.3: Solving the system of Equation 2.13, with 25 runs per row.

solving strategy	time **monodromy**	% $d = 36$	% $d = 88$	% $d = 114$
default	9.6 s	84 %	12 %	4 %
conservative tracker	11.9 s	100 %	0 %	0 %
w rank filter	10.9 s	100 %	0 %	0 %

2.4.2 The Mathieu Group M_{23}

To further demonstrate the versatility of `MonodromySolver`, we numerically verify some calculations due to Elkies [39] for a univariate family whose Galois/monodromy group is the Mathieu group M_{23}. This example illustrates some of the potential difficulties arising from naive application of homotopy continuation and monodromy as black-box methods, and some tricks for overcoming these difficulties.

Consider the univariate polynomials

$$\begin{aligned} P_2(x) = (8g^3 + 16g^2 - 20g + 20)x^2 - (7g^3 + 17g^2 - 7g + 76)x \\ + (-13g^3 + 25g^2 - 107g + 596), \end{aligned} \tag{2.14}$$

$$\begin{aligned} P_3(x) = 8(31g^3 + 405g^2 - 459g + 333)x^3 + (941g^3 + 1303g^2 - 1853g + 1772)x \\ + (85g^3 - 385g^2 + 395g - 220), \end{aligned} \tag{2.15}$$

$$\begin{aligned} P_4(x) = 32(4g^3 - 69g^2 + 74g - 49)x^4 + 32(21g^3 + 53g^2 - 68g + 58)x^3 \\ - 8(97g^3 + 95g^2 - 145g + 148)x^2 + 8(41g^3 - 89g^2 - g + 140)x \\ + (-123g^3 + 391g^2 - 93g + 3228), \end{aligned} \tag{2.16}$$

$$P(x) = P_2(x)^2\, P_3(x)\, P_4(x)^4, \tag{2.17}$$

where $g \approx .549472 - .67565i$ is an algebraic number satisfying

$$g^4 + g^3 + 9g^2 - 10g + 8 = 0. \tag{2.18}$$

The map $\mathbb{C} \ni x \mapsto P(x) \in \mathbb{C}$ is a degree-23 branched cover which is unramified over the twice-punctured plane $U = \mathbb{C} \setminus \{0, \tau\}$, where

$$\tau = (2^{38}3^{17}/23^3)(47323g^3 - 1084897g^2 + 7751g - 711002) \tag{2.19}$$

is a complex number whose modulus is on the order of 10^{21}. In principal, to compute the Galois/monodromy group of the branched cover $x \mapsto P(x)$ we only need to carefully track small loops around the ramification points 0 and τ. However, these equations are poorly-scaled—for instance, evaluation of $P(x)$ at random points in the complex unit circle typically results in complex numbers of modulus $\approx 10^{20}$, creating problems for most homotopy continuation software packages in their default settings.

To remedy the poor scaling, we define

$$\begin{aligned} Q_i(x) &= P_i(x)/10^3, \quad i = 2, 3, 4 \\ Q(x) &= Q_2(x)^2\, Q_3(x)\, Q_4(x)^4. \end{aligned} \tag{2.20}$$

With this normalization, the ramification points of $x \mapsto Q(x)$ become 0 and $\tau/10^{21} \approx -.165 + 2.29i$. Rather than tracking loops around these points, we perform an experiment that shows how we can compute the Galois/monodromy groups by tracking random loops in a higher-dimensional parameter space. We define a family of systems $f\left((X, Y, g); (a, b, c, d, e)\right)$ as follows:

$$\begin{aligned} f_1\left((X, Y, g); (a, b, c, d, e)\right) &= g^4 + g^3 + 9g^2 - 10g + 8 \\ f_2\left((X, Y, g); (a, b, c, d, e)\right) &= a\, Q(X/Y) + b \\ f_3\left((X, Y, g); (a, b, c, d, e)\right) &= c\, X + dY + e. \end{aligned} \tag{2.21}$$

Here, the last equation is a parametric chart on the projective line $\mathbb{P}^1$ with homogeneous coordinates $[X : Y] = [x : 1]$, and the first equation is chosen for the convenience

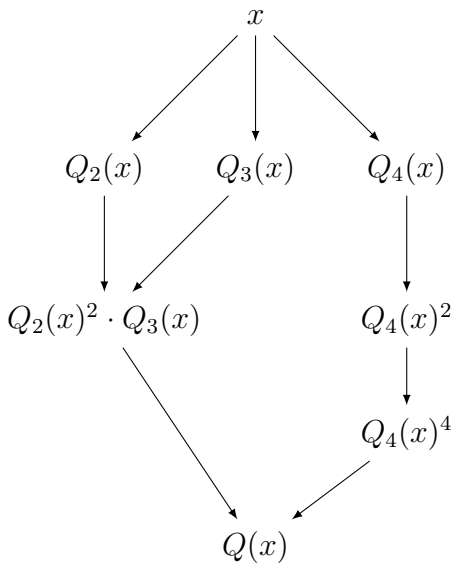


Figure 2.5: Schematic straight-line program that computes $Q(x)$.

of working with a system defined over $\mathbb{Q}$. In Equation 2.21 we have also emphasized that the second equation can be evaluated using straight-line programs implemented in the package `SLPexpressions`. The straight-line program for $Q(x)$ is depicted in Figure 2.5. This allows us to use fewer floating-point evaluations compared to Horner-like schemes used for polynomials in their dense coefficient representations. To seed the function `monodromySolve`, we fix $g_0 \approx .549472 - .67565i$, draw X_0, Y_0 randomly from the complex unit circle, and sample initial parameters $(a_0, b_0, c_0, d_0, e_0)$ by computing a random vector in the kernel of the matrix $d_{(a,\ldots,e)}\, f\,((X_0, Y_0);\, (a_0, \ldots, e_0)))$ via SVD. Despite better scaling, the problem remains poorly conditioned, so we use conservative tracking tolerances:

- `CorrectorTolerance => 1e-8`
- `tStepMin => 1e-12`
- `Precision => 100`
- `maxCorrSteps => 1`

With these settings, we ran Algorithm 2 on graphs with increasing Betti number. The non-seed vertices had parameters (a, b, c, d, e) drawn randomly from the complex unit circle.

Table 2.4: Estimated probability of generating M_{23} with random loops in $\mathbb{C}^5$.

$\#V(G)$	$\#E(G)$	$\beta_1(G)$	% M_{23}	% C_4
2	10	9	50 %	50 %
3	10	28	70 %	30 %
4	10	57	100 %	0 %

Edges were randomized by mapping each endpoint (a, b, c, d, e) to $(\gamma_1 a, \gamma_1 b, \gamma_2 c, \gamma_2 d, \gamma_2 e)$ for $\gamma_1,\ \gamma_2$ chosen randomly from the complex unit circle. Table 2.4 shows the results of 10 runs on a homotopy graph with $2,\ 3,\ 4$ vertices and 10 edges between every pair of them. Typical runs for 3 and 4 vertices took around 30 and 60 seconds, respectively. In all cases, the group returned by Algorithm 2 is typically either M_{23} or the cyclic group C_4 acting on only 4 discovered solutions. A uniform model of randomization suggests that 9 random elements of M_{23} generate the group with very high probability. Indeed, this can be checked by direct simulation in GAP [45]:

```
G:=MathieuGroup(23);
Length(Filtered(List([1..10000],i->Group(List([1..9], \
       i->Random(G)))=G), b -> b)); # typically 1000
```

Table 2.4 shows that the uniform model is very far off. Still, by increasing the Betti number we are gradually able to boost the chances of computing the whole group.

2.4.3 A problem from kinematics

We now move on to an optimization problem from kinematic design. In [14], the authors design planar mechanisms with polar linkages. For fixed values $l \in \mathbb{R}$ and $\mathbf{b}_1,\ \mathbf{b}_2,\ \mathbf{b}_3 \in \mathbb{R}^2$, the mechanism starting in an initial configuration $\mathbf{q} \in \mathbb{R}^2$ will sweep out a generator curve that meets certain technical design requirements. To enforce these design requirements, points $\mathbf{d}_1, \ldots, \mathbf{d}_N$ are sampled from the generator curve such that the *loop equations* $l - \|\mathbf{d}_j - \mathbf{R}_{k,j}\mathbf{q} - \mathbf{b}_k\|^2 = 0$ are all as close to zero as possible for $k \in \{1, 2, 3\}$ and $j \in \{1, \ldots, N\}$. For given parameters $(\mathbf{R}_{k,j})_{(1,1)}^{(3,N)}$ and $(\mathbf{d}_j)_{j=1}^{N}$, a mechanism that approximately

meets the design requirements can be found by optimizing the function

$$f\left(\mathbf{b}_1,\, \mathbf{b}_2,\, \mathbf{b}_3,\, \mathbf{q}, l; (\mathbf{R}_{k,j},\, \mathbf{d}_j\right) = \sum_{(k,j)=(1,1)}^{(3,N)} l - \|\mathbf{d}_j - \mathbf{R}_{k,j}\mathbf{q} - \mathbf{b}_k\|^2. \tag{2.22}$$

In the above, the notation $\|\cdot\|^2$ means the quadratic form on $\mathbb{C}^2$ that restricts to the usual norm on $\mathbb{R}^2$. To find the optimum for given parameters, it is sufficient to solve for all of the critical points, which satisfy a square system in the 9 unknowns:

$$d_{\mathbf{b}_1} f = d_{\mathbf{b}_2} f = d_{\mathbf{b}_3} f = f d_{\mathbf{q}} f = d_l f = 0. \tag{2.23}$$

This system, independently of N, has a BKK bound of $6,561$, and the authors report a root count $1,253$. We validate the reported root count by taking $N = 4$ and using `MonodromySolver`.

To do monodromy, the usual rational parametrization for the rotation parameters is a natural choice:

$$\mathbf{R}_{k,j}(t_{k,j}, u_{k,j}) = (t_{k,j}^2 + u_{k,j}^2)^{-1} \begin{pmatrix} t_{k,j}^2 - u_{k,j}^2 & -2t_{k,j}u_{k,j} \\ 2t_{k,j}u_{k,j} & t_{k,j}^2 - u_{k,j}^2 \end{pmatrix}. \tag{2.24}$$

Unfortunately, this led to unstable monodromy runs with the default settings—some nodes collected $1,257$ solutions, while others collected $1,253$. To explain this discrepency, we consider Equation 2.23 as parametrized by *un-normalized* rotations.

$$\mathbf{R}_{k,j}(t_{k,j}, u_{k,j}) = \begin{pmatrix} t_{k,j}^2 - u_{k,j}^2 & -2t_{k,j}u_{k,j} \\ 2t_{k,j}u_{k,j} & t_{k,j}^2 - u_{k,j}^2 \end{pmatrix}. \tag{2.25}$$

With this parametrization, we are able to reliably compute $1,257$ critical points on 4 vertices of a complete graph with no multiple edges, using the standard tracker settings. Each run takes about 10 minutes, in serial, which seems to compare favorably to the timing of 3 minutes real time with 192 cores for the case $N = 7$ reported in [14]. Once the monodromy run is finished, we run a parameter homotopy for Equation 2.23 with starting parameter values of the form in Equation 2.25 to target parameter values of the form

in Equation 2.24. We find that 4 paths seem to go off to infinity, bringing us into agreement with the results of [14]. To explain the discrepency from before, we note that rounding errors have the potential to make a system parametrized by Equation 2.24 look like a system parametrized by Equation 2.25. This leads to path-jumping much like we saw in Example 6, and underscores the fact that careful attention to the numerics may be needed for solving non-trivial problems like the system of Equation 2.23.

2.4.4 Pseudowitness sets

In numerical algebraic geometry, an irreducible variety $X \subset \mathbb{C}^n$ is represented by its intersection with a generic linear space L of complementary codimension. This is an effective representation of X in the sense that it allows for a *membership test*. To test a query of the form $x \in X$, one can use a homotopy $X \cap L \rightsquigarrow X \cap L_x$, where L_x is a genric linear space through x, and simply determine whether or not x is one of the endpoints reached from some point in $X \cap L$ (cf. [103, Ch. 13, 15] and [16, Ch. 8,16].) It is easy to implement such a homotopy method when equations vanishing on X are known—this leads to the notion of *witness sets*, which are the main data structures in numerical algebraic geometry. In other cases, we might only know X parametrically, say as the closed image of a rational map $X = \overline{\operatorname{im} \Psi}$ where $\Psi : \mathbb{C}^k \dashrightarrow \mathbb{C}^n$. The term *pseudowitness set* has been used in several previous works [27, 60, 61] to distinguish the parametric case.

Membership tests as described above can be applied to the invariant-theoretic problem of determining whether or not two algebraic curves are equivalent under some algebraic group action. This was the subject of the author's recent joint work with Ruddy [37]. The connection between pseudowitness sets and invariant theory is made through so-called *signature maps*. These signature maps originate from Cartan's method of moving frames [26] and have inspired a sizeable literature on the group-equivalence problem for smooth manifolds (see eg. [42].) More recently, signatures have been considered in the algebraic setting [69]. We describe two signature maps for the action of the Euclidean group $\mathbf{E}_{\mathbb{C}}(2) \curvearrowright \mathbb{C}^2$ by rotation and translation. If $C : F(x, y) = 0$ is an irreducible curve in $\mathbb{C}^2$, we let $y_C^{(1)} = y_x$, $y_C^{(2)} = y_{xx}, \ldots$

denote the implicit derivatives of y with respect to x. The map

$$\begin{aligned} \Psi_C : C &\dashrightarrow \mathbb{C}^2 \\ (x, y) &\mapsto \left(\frac{y_{xx}^2}{(1+y_x)^3}, \frac{\left(y_{xxx}(1+y_x)^2 - 3y_x y_{xx}\right)^2}{(1+y_x)^6} \right) \end{aligned}$$

is called the differential signature map associated to C. The coordinate functions of Ψ_C can be interpreted as the square of curvature and its derivative with respect to arc-length. The Zariski closure of the image of Ψ_C is called the differential signature, which we denote $\mathcal{S}_C$. The map is classifying in the sense that C_1 and C_2, besides certain exceptional cases, are $\mathrm{E}_{\mathbb{C}}(2)$-equivalent if and only if $\mathcal{S}_{C_1} = \mathcal{S}_{C_2}$—see [69, Theorem 2.37]. Thus, the homotopy membership test can be applied, in principal, to get a probabilistic test of $\mathrm{E}_{\mathbb{C}}(2)$-equivalence of curves—similarly for other matrix groups acting linearly on $\mathbb{C}^2$, with the appropriate signature map. To implement this test, we need to compute a pseudowitness set for one of the $\mathcal{S}_{C_i}$. This is where monodromy comes into play—by moving the blue line in Figure 2.6 in and out of place with monodromy loops, we can compute all of the points in the pseudowitness set for the signature curve. Stopping criteria are quite easy for generic C of degree d, since we have $\deg \mathcal{S}_C = 12\, d(d-1)$ [69, Theorem 4.13]. Sample timings for computing these pseudowitness sets are in Figure 2.6. We note that, for C of degree d, the rational functions defining Ψ_C typically involve polynomials of degree $6(d-2)$. Already for $d = 4, 5, 6$, this bodes poorly for the numerics if these rational functions are evaluated naively. Fortunately, there is a natural straight-line program structure as in the Example of subsection 2.4.2, thanks to the following recursion:

$$y_C^{(1)} = \frac{-\partial_x F}{\partial_y F} \quad \text{and} \quad y_C^{(k+1)} = \partial_x\, y_C^{(k)} + \partial_y\, y_C^{(k)}\, y_C^{(1)}.$$

Another type of signature map is the so-called *Euclidean joint signature map*

$$\begin{aligned} C \times C \times C \times C &\mapsto \mathbb{C}^6 \\ ((x_i, y_i))_{i=1}^4 &\mapsto \left((x_i - x_j)^2 + (y_i - y_j)^2\right)_{1 \leqslant i < j \leqslant 4} \end{aligned}$$

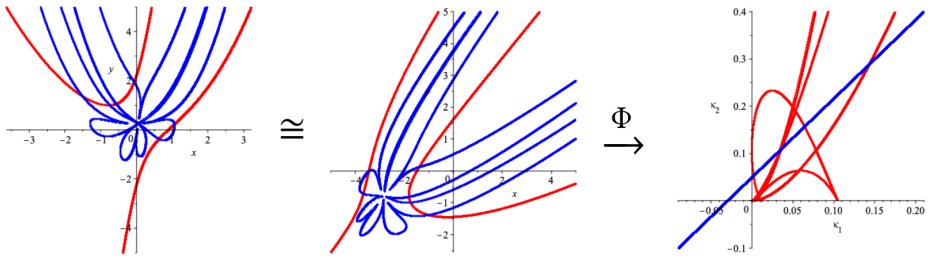


Figure 2.6: Two $\mathrm{E}_{\mathbb{C}}(2)$-equivalent curves and their differential signature in red. Where red meets blue, we get the points defining a witness set for the signature curve.

d	$\deg \mathcal{S}$	time (s)	$\deg \mathcal{J}$	time (s)	$\deg_{\boldsymbol{e_1}} \mathcal{J}$	time (s)	$\deg_{\boldsymbol{e_2}} \mathcal{J}$	time (s)
2	6	0.3	42	4	24	2	26	2
3	72	2	936	33	576	17	696	16
4	144	9	3024	139	1920	57	2448	87
5	240	21	7440	463	4800	206	6320	276
6	360	55	15480	1315	10080	748	13560	791

Figure 2.7: Degrees and monodromy timings for differential and joint signatures.

whose closed image $\mathcal{J}_C$ we call the *joint signature*. The Euclidean joint signature is typically a four-fold, and like the differential signature it is classifying for the group action. Equations of the joint signature map are much less complicated, but the size of witness sets are much larger. Figure 2.7 gives degrees and monodromy timings for the witness sets of the generic joint signature. We also considered *multiprojective* witness sets [58, 56], obtained by coordinates slices fixing some $d_{i,j}$. There are two combinatorial types, indexed by lattice vectors $\boldsymbol{e}_1 = (1,1,1,1,0,0)$ and $\boldsymbol{e}_2 = (0,1,1,1,1,0)$ inside of a certain polymatroid base polytope (see [56, Sec. 2] for details.) These computations allowed the authors of [37] to compare differential and joint signatures experimentally for the application to group equivalence described above. They also led to the following conjecture:

Conjecture 2.4.1. Let $\mathscr{I}_d$ denote the Euclidean joint signature for a generic plane curve of degree d. For $d \geqslant 3$:

$$\begin{aligned} \deg \mathscr{I}_d &= 12d(d^3-1) \\ \deg_{\boldsymbol{e}_1} \mathscr{I}_d &= 8d^2(d^2-1) \\ \deg_{\boldsymbol{e}_2} \mathscr{I}_d &= 4d(d-1)(3d^2+d-1). \end{aligned}$$

CHAPTER 3
VISION

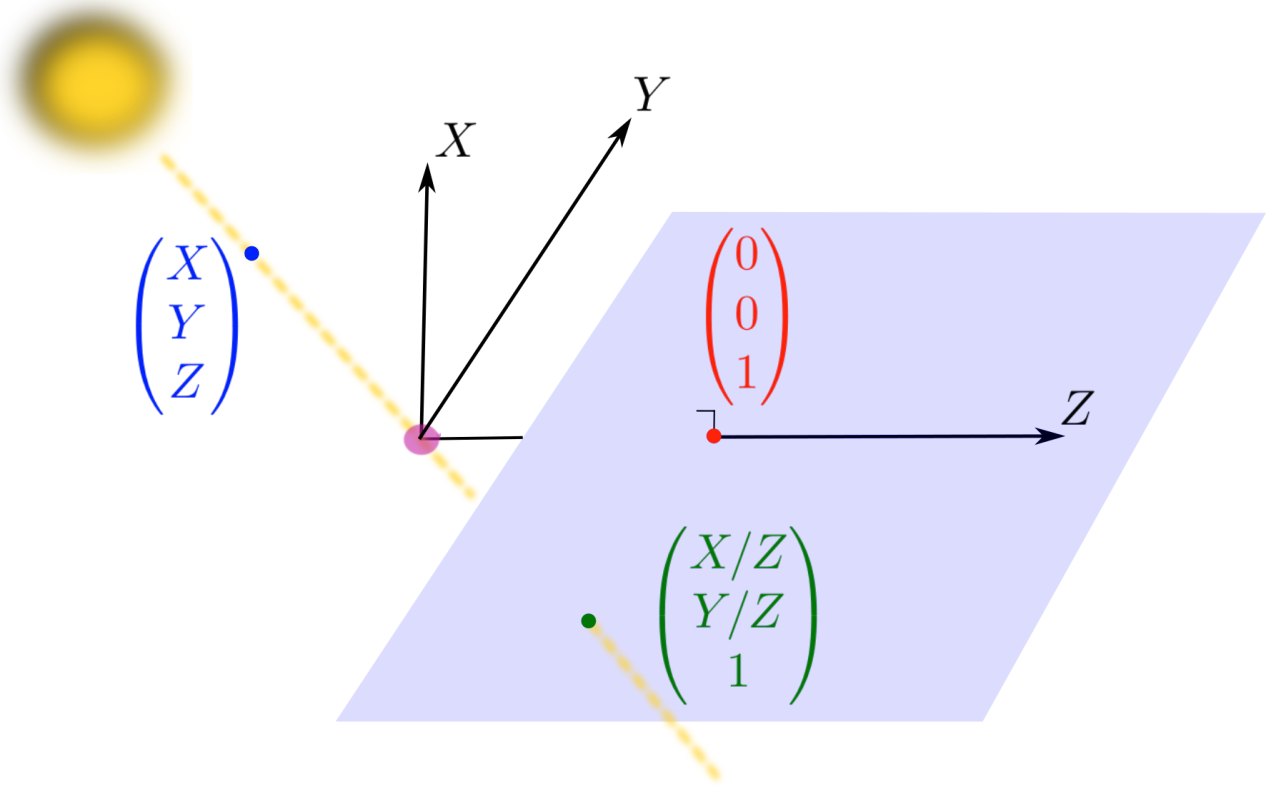


Figure 3.1: Digram of a pinhole camera with principal point $(0, 0)$ and focal length 1.

In this chapter, we apply the framework developed in the previous chapter to novel problems originating from the subject of *multiview geometry*. This is a sub-field of computer vision which studies the contraints placed on some number of *cameras* that see objects in a 3D world, as well as algorithms for recovering the cameras and 3D geometry. We give a few definitions and a condensed overview of the subject in section 3.1, giving particular attention to *minimal problems* which appear in RANSAC-based reconstruction. In section 3.2 and section 3.3, we summarize the main results of the papers [34] and [35]. We also spell out some of the nitty-gritty in the monodromy computations done in these works.

3.1 Preliminary notions

3.1.1 What is a camera?

We begin with a rapid summary of concepts in geometric computer vision. More details can be found in several standard texts [53, 85, 83]. Our starting point is the pinhole camera model, illustrated in Figure 3.1. Here, a real-life camera is modeled as projection through a point in space onto a plane. The center of projection is an idealized lens, through which rays of light pass to form an image. The coordinates of our *camera frame* are chosen so that the center of projection is the origin and so that the *image plane* is given by $H = \{(X, Y, Z) \in \mathbb{R}^3 \mid Z = 1\}$. We remark that our choice of coordinates implies that two world points which differ by sign $(X, Y, Z), (-X, -Y, -Z) \in \mathbb{R}^3$ will produce the same image in $\mathbb{R}^2$.

The equations of a world-to-image map for this camera, which can be derived algebraically or using similar triangles, are as follows:

$$\begin{aligned} \mathbb{R}^3 &\dashrightarrow H \\ (X, Y, Z) &\mapsto (X/Y, Y/Z, 1) \end{aligned} \tag{3.1}$$

This map is non-linear, and undefined when $Z = 0$. However, it can be better understood through the lens of *projective geometry.* In this approach, each point in the image is naturally identified with the light-ray that passes through it—-a line through the origin in $\mathbb{R}^3$. There are also exceptional light-rays where the map Equation 3.1 is undefined. These exceptional lines in space correspond to “vanishing points" where two parallel lines in our image meet. The projective space $\mathbb{P}^2$ is the space of *all* lines through the origin in $\mathbb{R}^3$. Using *homogeneous coordinates* on $\mathbb{P}^2$, we may rewrite our image coordinates as $[X/Y : X/Z : 1] = [X : Y : Z]$. In doing so, we obtain a *projective-linear* map from the world to the image

$$\begin{aligned} \mathbb{P}^3 &\dashrightarrow \mathbb{P}^2 \\ [X : Y : Z : W] &\mapsto [X : Y : Z] \end{aligned} \tag{3.2}$$

which is defined for all world points except the camera center $[0:0:0:1]$. Equivalently, the map is given by the camera matrix

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}.$$

A more general camera matrix has the form $\mathbf{K}[\mathbf{R} \mid \mathbf{t}]$ where

$$\mathbf{K} = \begin{pmatrix} \alpha_x & s & x_0 \\ 0 & \alpha_y & y_0 \\ 0 & 0 & 1 \end{pmatrix}$$

is the so-called calibration matrix and $[\mathbf{R} \mid \mathbf{t}] \in \mathrm{SE}_{\mathbb{R}}(3)$ is a 3×4 matrix in the special Euclidean group giving the relative orientation between the world and camera frame. The parameters can be understood as follows: α_x and α_y measure the focal length (that is, the distance from the camera center to the plane) in terms of the pixel dimensions in x and y directions of the image, x_0 and y_0 give the principal point (the center of the image), and s is a skew parameter. Almost any 3×4 matrix can be written as $\mathbf{K}[\mathbf{R} \mid \mathbf{t}]$ up to scale. In this chapter, we consider problems where the calibration matrix is known *a priori* and invertible.

3.1.2 What is a minimal problem?

In various applications, the information provided by *multiple* cameras must be combined. These applications may involve very intensive computation (eg. building 3D models of cities from large collections of photos [1]) or have strict real-time requirements (eg. multi-camera systems for 360° field-of-view on an autonomous vehicle [51].) A common thread throughout these applications is that the data in images are noisy. Even worse, the heuristic nature of algorithms for matching features between images can result in non-negligible amount of mislabeled data, commonly known as *outliers.* For example, a common task is to estimate the relative orientation between two cameras given many point correspondences between pairs of corresponding points $(\mathbf{x}_1, \mathbf{y}_1), \ldots, (\mathbf{x}_m, \mathbf{y}_m) \in \mathbb{P}^2_{\mathbb{R}} \times \mathbb{P}^2_{\mathbb{R}}$. If the calibration

matrices for both cameras are known, we may make the simplifying assumption that the frame of the first camera is equal to the world frame, and normalize each image points by pre-multiplying by the inverse of its calibration matrix. Thinking of each $\mathbf{x}_i, \mathbf{y}_i$ as a 3×1 matrix with last coordinate $= 1$ (ie. in the affine chart $Z = 1$)), we get point correspondence equations

$$\beta_i \, \mathbf{y}_i = \mathbf{R} \, (\alpha_i \mathbf{x}_i) + \mathbf{t} \tag{3.3}$$

where $\alpha_1, \ldots, \alpha_m, \beta_1, \ldots, \beta_m$ are the the unknown depths of the points in the first and second camera. For this problem, outliers are mismatched pairs $(\mathbf{x}_i, \mathbf{y}_i)$ which do not correspond to a common world point seen by both cameras.

The prevalence of outliers calls for robust estimation techniques, among which heuristics based on *RANSAC* (RANdom SAmpling and Consensus) [43] have been ubiquitous in computer vision. The basic idea of RANSAC is to sample m point correspondences at random and solve Equation 3.3 for $[\mathbf{R} \mid \mathbf{t}] \in \mathrm{SO}_{\mathbb{R}}(3)$ until an outlier-free subset is detected by comparison against the rest of the data. Suppose that $w \in [0, 1]$ is the fraction of inlier correspondences; how long must we wait to get an outlier-free sample with probability $\delta = .95$? Since, the probability of waiting $\geqslant k$ trials is $(1 - w^{-m})^k$, we may expect

$$k = \log(1 - \delta) / \log(1 - w^{-m}) \approx 3 \, w^{-m}$$

iterations are needed. The implications for RANSAC are clear: the sample-size m should be as small as possible, and a *minimal solver* for Equation 3.3 should run as fast as possible. These needs are amplified within structure-from-motion systems, which use many RANSAC runs to obtain initial pose estimates between many cameras, which are later refined with nonlinear least-squares (see eg. [97, 1, 100, 11].)

Minimal problems and minimal solvers are an active area of research in computer vision—see eg. [71, 75, 76, 72, 4]. Many minimal solvers are constructed via an offline/online phase analogous to the use of parameters homotopies described in the introduction. However, most minimal solvers are based on Gröbner bases and resultants; for use in RANSAC, the target runtime of a minimal solver is typically on the order of micro-seconds. Despite their successes, these techniques are typically limited to problems of low degree (< 100.)

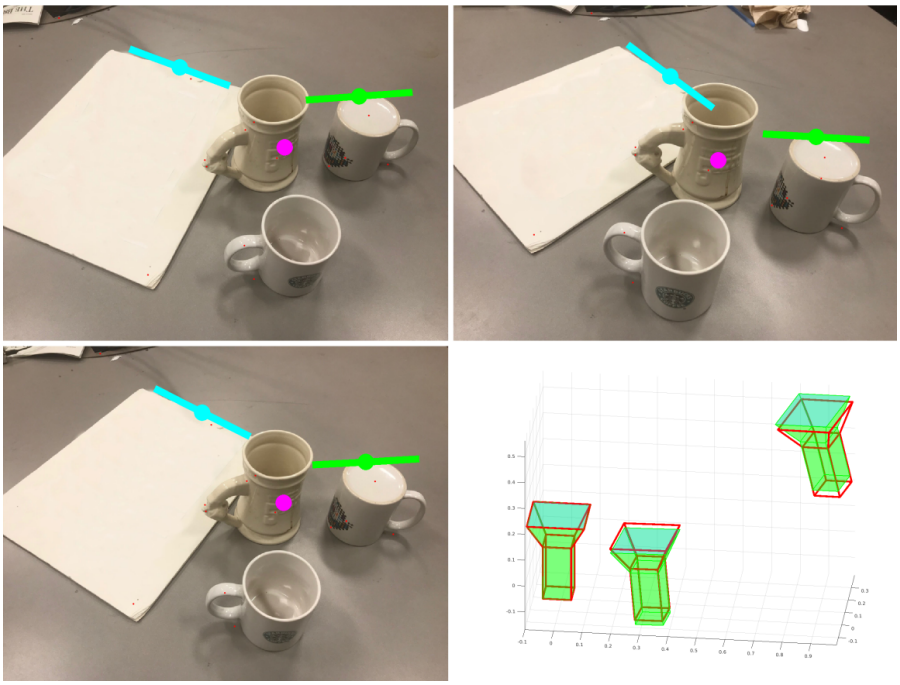


Figure 3.2: Recovering cameras from point-line incidences, taken from [40].

In work with Fabbri et al. [40], the author helped to develop a homotopy-based minimal solver for a novel minimal problem of degree 312. This problem, nicknamed the "Chicago problem", is motivated by the failure of structure-from-motion pipelines on textureless scenes where few point-point correspondences are available, such as in Figure 3.2. Feature detectors such as SIFT (Scale-Invariant Feature Transform [82]) are designed to detect both a point and an orientation in each image. The SIFT orientation is effectively a line incident to the detected point, and a line-line-line correspondence between three cameras constrains the 3D geometry.

Although too slow for immediate use in RANSAC ($\approx$ 1s per run), the solver developed for the Chicago problem showed that homotopy continuation is a promising tool for solving solving minimal problems of higher degree and with more than two cameras. The offline solver used was `MonodromySolver`, and the formulation for the final solver is the same as will be discussed for the other *point-line minimal problems* treated in this chapter.

3.2 Point-line minimal problems with complete visibility

In this subsection, we classify minimal problems for relative pose estimation in which a configuration of points and lines is observed by any number of calibrated cameras. There are 30 such minimal problems for $m = 2, 3, 4, 5$, or 6 cameras, subject to the assumption that every entity (point, line, or incidence) that occurs in the world is seen in every camera. We call this assumption *complete visibility.* We compute the degrees of such problems for $m = 2, 3, 4$, and 5 using monodromy. For $m = 2, 3$, these computations agree with the answers obtained by symbolic computation (Gröbner bases.) In fact, the monodromy runs helped us discover a bug in an older version of the symbolic code which resulted in incorrect degrees for certain problems.

The *point-line problems* we classify share a common encoding, $(p, l, \mathcal{I}, m)$, where

- p denotes the number of world points observed in each image
- l denotes the number of lines observed in each image
- $\mathcal{I} \subset \{1, \ldots, p\} \times \{1, \ldots, l\}$ is an incidence relation which must be satisfied for all points and lines in the world (and hence also in the images.) Points and lines in each image are indexed so that, for cameras P_v and $P_{v'}$, every pair of corresponding points that they see has the form $x_{v,i}, x_{v',i}$, (respectively in the case of lines: $\ell_{v,j}, \ell_{v',j}$.) Thus, if $(i, j) \in \mathcal{I}$, this means that $x_{v,i} \in \ell_{v,j}$ for all $v = 1, \ldots m$, where m is the number of cameras/images.

In this setup, we may easily model lines in space which intersect by requiring that each intersection point of two lines has to be one of the p points in the point-line problem. We shall assume that the incidence relation $\mathcal{I}$ is *realizable* as the set of incidences of some collection of points and lines in $\mathbb{P}^3$. Thus, two distinct lines cannot be incident to the same two distinct points. In addition, we will always assume that the incidence relation $\mathcal{I}$ is complete in the sense that every incidence which is automatically implied by the incidences in $\mathcal{I}$ must also be contained in $\mathcal{I}$.

A point-line arrangement in space consists of p points $X_1, \ldots, X_p$ and l lines $L_1, \ldots, L_l$ in $\mathbb{P}^3$ which are incident exactly as specified by $\mathcal{I} \subset \{1, \ldots, p\} \times \{1, \ldots, l\}$. Hence, the

point X_i is on the line L_j if and only if $(i,j) \in \mathcal{I}$. We write

$$\mathcal{X}_{p,l,\mathcal{I}} = \left\{ (X, L) \in \left(\mathbb{P}^3\right)^p \times \left(\mathbb{G}_{1,3}\right)^l \mid \forall (i,j) \in \mathcal{I} : X_i \in L_j \right\}$$

for the associated *variety of point-line arrangements*. Note that this variety also contains degenerate arrangements, where not all points and lines have to be pairwise distinct or where there are more incidences between points and lines than those specified by $\mathcal{I}$.

Let us now consider a generic point-line arrangement in $\mathcal{X}_{p,l,\mathcal{I}}$. We partition the set of points into independent and depdendent points, where a dependent point lies on a line spanned by two other points, such that the number of independent points is minimal. We write p^{f} and $p^{\mathrm{d}} = p - p^{\mathrm{f}}$ for the number of independent and dependent points, respectively (the upper index f stands for *free*). Each free point is defined by three parameters. A dependent point X is only defined by *one* further parameter after the two points, which span the line containing X, are defined. In total, the p points in our arrangement are defined by $3\,p^{\mathrm{f}} + p^{\mathrm{d}}$ parameters. Each of the l lines in our arrangement is either incident to zero, one or at least two points. We refer to lines which are incident to no points as *free lines*. We denote the number of free lines by l^{f}. As the Grassmannian $\mathbb{G}_{1,3}$ of lines is four-dimensional, each free line is defined by four parameters. A line which is incident to a fixed point is defined by only two parameters. We denote the number of lines which are incident to exactly one point by l^{a} (the upper index a stands for *adjacent*). Finally, each of the remaining $l - l^{\mathrm{f}} - l^{\mathrm{a}}$ lines is incident to at least two points and thus already uniquely determined by the two points. Hence, we have derived

$$\dim(\mathcal{X}_{p,l,\mathcal{I}}) = 3\,p^{\mathrm{f}} + p^{\mathrm{d}} + 4\,l^{\mathrm{f}} + 2\,l^{\mathrm{a}}. \tag{3.4}$$

In particular, we see that we might as well assume that there is no line passing through two or more points, as such lines do not contribute to our dimension count. Indeed, we would get no new minimal problems up to birational equivalence if we relaxed this condition. We also note that $\mathcal{X}_{p,l,\mathcal{I}}$ admits a rational parametrization and thus is irreducible.

The space of inputs/data for a point-line problem is defined analagously to $\mathcal{X}_{p,l,\mathcal{I}}$:

$$\mathcal{Y}_{p,l,\mathcal{I},m} = \left\{ (x, \ell) \in \left(\mathbb{P}^2\right)^{m\,p} \times \left(\mathbb{G}_{1,2}\right)^{m\,l} \;\middle|\; \forall v = 1, \ldots, m \; \forall (i,j) \in \mathcal{I} : x_{v,i} \in \ell_{v,j} \right\}$$

for the *image variety*, which consists of all m-tuples of two-dimensional point-line arrangements which satisfy the incidences specified by $\mathcal{I}$.

We derive the dimension of the image variety $\mathcal{Y}_{p,l,\mathcal{I},m}$ similarly. Since we assume all camera positions to be sufficiently generic, each camera views exactly p^{f} independent points, p^{d} dependent points, l^{f} free lines and l^{a} lines which are incident to exactly one of the points. Each independent point is defined by two parameters, whereas each dependent point is defined by a single parameter. A free line is defined by two parameters. A line which is incident to a fixed point is defined by a single parameter. All in all, we have that

$$\dim(\mathcal{Y}_{p,l,\mathcal{I},m}) = m\,(2\,p^{\mathrm{f}} + p^{\mathrm{d}} + 2\,l^{\mathrm{f}} + l^{\mathrm{a}}). \tag{3.5}$$

There are two inherent ambiguities in recovering calibrated camera matrices $[\mathbf{R}_1 \mid \mathbf{t}_1], \ldots, [\mathbf{R}_m \mid \mathbf{t}_m] \in \mathrm{SE}_{\mathbb{C}}(3)$ from data in images. The first is arbitrary choice of a Euclidean coordinate system for the world $\mathbb{C}^3$. The second is the ability to "re-scale the world": if $X_i \in \mathbb{C}^3$ and $x_i \in \mathbb{C}^2$ are such that

$$\begin{pmatrix} \mathbf{R}_k & | & \mathbf{t}_k \end{pmatrix} \begin{pmatrix} X_i \\ 1 \end{pmatrix} \sim \begin{pmatrix} x_i \\ 1 \end{pmatrix}, \qquad k = 1, \ldots, m,$$

then also for any $c \in \mathbb{C}$ we have

$$\begin{pmatrix} \mathbf{R}_k & | & c\,\mathbf{t}_k \end{pmatrix} \begin{pmatrix} c\,X_i \\ 1 \end{pmatrix} \sim \begin{pmatrix} x_i \\ 1 \end{pmatrix}, \qquad k = 1, \ldots, m,$$

To fix the first ambiguity, we assume the world frame and the frame of the first camera coincide. To fix the second ambiguity, we may assume for generic data that the first

coordinate of $\mathbf{t}_2$ equals 1. Thus, we define the space of calibrated camera configurations as

$$\mathcal{C}_m = \left\{ (P_1, \ldots, P_m) \in \mathrm{SE}_{\mathbb{C}}(3) \;\middle|\; P_1 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}, P_2 = \begin{pmatrix} * & * & * & 1 \\ * & * & * & * \\ * & * & * & * \end{pmatrix} \right\}. \quad (3.6)$$

We have quite easily that $\dim \mathcal{C}_m = 6m - 7$.

The *joint camera map* $\Phi_{p,l,\mathcal{I},m} : \mathcal{X}_{p,l,\mathcal{I}} \times \mathcal{C}_m \dashrightarrow \mathcal{Y}_{p,l,\mathcal{I},m}$ sends a point-line arrangement in space and m cameras to the resulting joint image.

Definition 3.2.1. We say that the point-line problem $(p, l, \mathcal{I}, m)$ is *minimal* if the joint camera map $\Phi_{p,l,\mathcal{I},m} : \mathcal{X}_{p,l,\mathcal{I}} \times \mathcal{C}_m \dashrightarrow \mathcal{Y}_{p,l,\mathcal{I},m}$ is a branched cover.

The degree of the minimal problem specified by $(p, l, \mathcal{I}, m)$ is defined to be the degree of its joint camera map. In order to be a minimal problem, it is necessary that the 3D degrees of freedom be equal to the number of constraints implied by the 2D data. More precisely, Definition 2.1.1 requires that the tuple $(p, l, \mathcal{I}, m)$ be *balanced* in the following sense.

Definition 3.2.2. $(p, l, \mathcal{I}, m)$ is balanced if $\dim(\mathcal{X}_{p,l,\mathcal{I}} \times \mathcal{C}_m) = \dim(\mathcal{Y}_{p,l,\mathcal{I},m})$.

We can easily list all balanced point-line problems for any particular m by counting dimensions. As it turns out, there are only finitely many balanced problems for *all* m. These are given explicitly in Table 3.1, up to the addition of an arbitrary number of line correspondences for $m = 2$ (see Remark 3.2.6.)

To weed out the non-minimal balanced problems, it is enough to compute the Jacobian of $\Phi_{p,l,\mathcal{I},m}$ in local coordinates around a generic $(x, c) \in \mathcal{X}_{p,l,\mathcal{I}} \times \mathcal{C}_m$ and check whether or not it is invertible for each problem $(p, l, \mathcal{I}, m)$. As a result, it is easy to check which of the balanced problems appearing in Table 3.1 are actually minimal.

3.2.1 Balanced Point-Line Problems

To enumerate balanced problems, we use the dimension counts from the previous section. Note that there is no balanced point-line problem for a single camera. For $m > 1$ cameras, combining $\dim(\mathcal{C}_m) = 6m - 7$ with Equation 3.4 and Equation 3.5 yields that a point-line

Table 3.1: All balanced point-line problems, modulo adding arbitrarily many lines to the problems with 2 views. Each picture gives the combinatorial type of point-line incidences $\mathcal{I}$ in both the world and in image correspondences. Degrees are listed for the minimal problems. Degrees marked with * were only computed using monodromy, whereas the others were verified using both monodromy and Gröbner bases.

6	6	6	5	5	5	4	4	4	4	4	4	4
												
$> 180k^*$			11296*	26240*	11008*	3040*	4512*			1728*	32*	544*
4	3	3	3	3	3	3	3	3	3	3	3	3
												
544*	360	552	480			264	432	328	480	240	64	216
3	3	3	3	3	3	3	3	2	2	2	2	2
												
312	224	40	144	144	144	64		20	16	12		

problem is balanced if and only if

$$3\,p^{\mathrm{f}} + p^{\mathrm{d}} + 4\,l^{\mathrm{f}} + 2\,l^{\mathrm{a}} + 6\,m - 7 = m\left(2\,p^{\mathrm{f}} + p^{\mathrm{d}} + 2\,l^{\mathrm{f}} + l^{\mathrm{a}}\right).$$

This is equivalent to

$$6m-7=(2m-3)p^{\mathrm{f}}+(m-1)p^{\mathrm{d}}+2(m-2)l^{\mathrm{f}}+(m-2)l^{\mathrm{a}}. \tag{3.7}$$

Lemma 3.2.3. Every balanced point-line problem with at least five points has exactly two cameras.

Proof. Suppose $(p, l, \mathcal{I}, m)$ is a balanced point-line problem with $m > 1$ cameras and at least five points, i.e. $p^{\mathrm{f}} + p^{\mathrm{d}} \geqslant 5$. In this case, the equality Equation 3.7 implies

$$6m-7 \geqslant (2m-3)p^{\mathrm{f}}+(m-1)(5-p^{\mathrm{f}})=(p^{\mathrm{f}}+5)m-(2p^{\mathrm{f}}+5),$$

which is equivalent to

$$2(p^{\mathrm{f}}-1) \geqslant (p^{\mathrm{f}}-1)m. \tag{3.8}$$

Among the five or more points at least two have to be (by definition) independent, i.e. $p^{\mathrm{f}} > 1$. So Equation 3.8 yields $m \leqslant 2$. □

Theorem 3.2.4. There is no balanced point-line problem with seven or more cameras.

Proof. Let $(p, l, \mathcal{I}, m)$ be a balanced point-line problem with $m \geqslant 7$ cameras. By equality Equation 3.7, we have

$$5 \equiv p^{\mathrm{f}} + p^{\mathrm{d}} \mod (m-2). \tag{3.9}$$

This implies $p^{\mathrm{f}} + p^{\mathrm{d}} \geqslant 5$ if $m \geqslant 8$, which contradicts Lemma 3.2.3, and thus we have only one remaining case to check: $m = 7$. From Equation 3.9 and Lemma 3.2.3, we have $p^{\mathrm{f}} + p^{\mathrm{d}} = 0$ in the case of seven cameras. It means that there are no points, and thus there cannot be lines which are incident to points. So we have $p^{\mathrm{f}} = 0$, $p^{\mathrm{d}} = 0$, $l^{\mathrm{a}} = 0$, and Equation 3.7 reduces to $35 = 10l^{\mathrm{f}}$, which is clearly not possible. So there are no balanced point-line problems with seven or more cameras. □

Theorem 3.2.5. There are 34 balanced point-line problems with 3, 4, 5 or 6 cameras. They are all listed in Table 3.1.

Proof. We consider the different cases for $3 \leqslant m \leqslant 6$ and reason by cases.

• $m = 6$: Due to Equation 3.9 and Lemma 3.2.3, every balanced point-line problem with six cameras must have exactly one point. So we have $p^{\mathrm{f}} = 1$, $p^{\mathrm{d}} = 0$, and Equation 3.7 reduces to $5 = 2l^{\mathrm{f}} + l^{\mathrm{a}}$. This gives us three possibilities: $(l^{\mathrm{f}}, l^{\mathrm{a}}) \in \{(2,1), (1,3), (0,5)\}$ (see first row of Table 3.1).

• $m = 5$: Due to Equation 3.9 and Lemma 3.2.3, every balanced point-line problem with five cameras must have exactly two points. So we have $p^{\mathrm{f}} = 2$, $p^{\mathrm{d}} = 0$, and Equation 3.7 reduces to $3 = 2l^{\mathrm{f}} + l^{\mathrm{a}}$. This gives us two possibilities: $(l^{\mathrm{f}}, l^{\mathrm{a}}) \in \{(1,1), (0,3)\}$, which yield three point-line problems (see the first row of Table 3.1).

• $m = 4$: Due to Equation 3.9 and Lemma 3.2.3, every balanced point-line problem with four cameras must have either one point or three points. Let us first consider the case of a single point. Here we have $p^{\mathrm{f}} = 1$, $p^{\mathrm{d}} = 0$, and Equation 3.7 reduces to $6 = 2l^{\mathrm{f}} + l^{\mathrm{a}}$. This gives us four possibilities: $(l^{\mathrm{f}}, l^{\mathrm{a}}) \in \{(3,0), (2,2), (1,4), (0,6)\}$ (see first

row of Table 3.1). Secondly, we consider balanced point-line problems with four cameras and three points. If all three points are independent, Equation 3.7 reduces to $1 = 2l^{\mathrm{f}} + l^{\mathrm{a}}$, which has a single solution: $(l^{\mathrm{f}}, l^{\mathrm{a}}) = (0, 1)$. If not all three points are independent, we have $p^{\mathrm{f}} = 2$, $p^{\mathrm{d}} = 1$, and Equation 3.7 reduces to $2 = 2l^{\mathrm{f}} + l^{\mathrm{a}}$. This gives us two possibilities: $(l^{\mathrm{f}}, l^{\mathrm{a}}) \in \{(1, 0), (0, 2)\}$, which yield three point-line problems (see the first two rows of Table 3.1 for all four point-line problems with four cameras and three points).

• $m = 3$: We first observe that each balanced point-line problem with three cameras must have at least one point. Otherwise we would have $p^{\mathrm{f}} = 0$, $p^{\mathrm{d}} = 0$ and $l^{\mathrm{a}} = 0$, so Equation 3.7 would reduce to $11 = 2l^{\mathrm{f}}$, which is impossible. Let us first consider the case of a single point. Here we have $p^{\mathrm{f}} = 1$, $p^{\mathrm{d}} = 0$, and Equation 3.7 reduces to $8 = 2l^{\mathrm{f}} + l^{\mathrm{a}}$. This gives us five possibilities: $(l^{\mathrm{f}}, l^{\mathrm{a}}) \in \{(4, 0), (3, 2), (2, 4), (1, 6), (0, 8)\}$ (see second row of Table 3.1). Secondly, in the case of two points, we have $p^{\mathrm{f}} = 2$, $p^{\mathrm{d}} = 0$, and Equation 3.7 reduces to $5 = 2l^{\mathrm{f}} + l^{\mathrm{a}}$. This gives us three possibilities: $(l^{\mathrm{f}}, l^{\mathrm{a}}) \in \{(2, 1), (1, 3), (0, 5)\}$, which yield six point-line problems (see second row of Table 3.1). Thirdly, we consider the case of three points. If all three points are independent, Equation 3.7 reduces to $2 = 2l^{\mathrm{f}} + l^{\mathrm{a}}$. The two solutions $(l^{\mathrm{f}}, l^{\mathrm{a}}) \in \{(1, 0), (0, 2)\}$ yield three point line problems (see last two rows of Table 3.1). If not all three points are independent, we have $p^{\mathrm{f}} = 2$, $p^{\mathrm{d}} = 1$, and Equation 3.7 reduces to $3 = 2l^{\mathrm{f}} + l^{\mathrm{a}}$. The two solutions $(l^{\mathrm{f}}, l^{\mathrm{a}}) \in \{(1, 1), (0, 3)\}$ yield four point-line problems (see last row of Table 3.1). Finally, we consider balanced point-line problems with three cameras and four points. We see from Equation 3.7 that not all four points can be independent. Hence, we either have $p^{\mathrm{f}} = 3$ and $p^{\mathrm{d}} = 1$ such that Equation 3.7 reduces to $0 = 2l^{\mathrm{f}} + l^{\mathrm{a}}$, which has a single solution $(l^{\mathrm{f}}, l^{\mathrm{a}}) = (0, 0)$, or we have $p^{\mathrm{f}} = 2$ and $p^{\mathrm{d}} = 2$ such that Equation 3.7 reduces to $1 = 2l^{\mathrm{f}} + l^{\mathrm{a}}$, which also has a single solution $(l^{\mathrm{f}}, l^{\mathrm{a}}) = (0, 1)$ (see the last row of Table 3.1) □

Remark 3.2.6. For the case of two cameras, we see from Equation 3.7 that the number of free and incident lines do not contribute to the dimension count for balanced point-line problems. In fact, Equation 3.7 reduces for $m = 2$ to $5 = p^{\mathrm{f}} + p^{\mathrm{d}}$. Hence, we have the classical minimal problem of recovering five points from two camera images. More precisely, a point-line problem with two cameras is balanced if and only if it has five points. Therefore, it is irrelevant how many lines are contained in the arrangement or how many

points are independent. There are 5 combinatorial possibilities to distribute dependent and independent points (see the last row of Table 3.1).

Corollary 3.2.7. There are 39 balanced point-line problems, modulo any number of lines in the case of two views. They are listed in Table 3.1.

3.2.2 Eliminating world points and lines

Having reduced the classification of minimal problems to finitely many candidates, it remains to check which balanced point-line problems are minimal. In order to do computations, it is customary to describe problems with implicit equations that do not depend on the world variables. Before we describe such equations, let us phrase the elimination of the world variables geometrically.

We consider the *graph* of the joint camera map: that is, the incidence variety

$$\mathrm{Inc} = \{(X, C, Y) \in \mathcal{X}_{p,l,\mathcal{I}} \times \mathcal{C}_m \times \mathcal{Y}_{p,l,\mathcal{I},m} \mid Y = \Phi_{p,l,\mathcal{I},m}(X, C)\}.$$

The joint camera map $\Phi_{p,l,\mathcal{I},m}$ is birationally equivalent to the map $\pi_{\mathcal{Y}} : \mathrm{Inc} \to \mathcal{Y}_{p,l,\mathcal{I},m}$ obtained by projection onto the last factor. We may also consider a restricted incidence variety which does not include the 3D structure $\mathcal{X}_{p,l,\mathcal{I}}$:

$$\mathrm{Inc}' = \{(C, Y) \in \mathcal{C}_m \times \mathcal{Y}_{p,l,\mathcal{I},m} \mid \exists X \in \mathcal{X}_{p,l,\mathcal{I}} : Y = \Phi_{p,l,\mathcal{I},m}(X, C)\}.$$

We have a diagram

$$\begin{array}{ccc} \mathrm{Inc} & \xrightarrow{\pi_{\mathcal{Y}}} & \mathcal{Y}_{p,l,\mathcal{I},m} \\ {\scriptstyle \pi_{\mathcal{C},\mathcal{Y}}}\downarrow & \nearrow {\scriptstyle \pi'_{\mathcal{Y}}} & \\ \mathrm{Inc}' & & \end{array}$$

where $\pi_{\mathcal{C},\mathcal{Y}}$ omits the first factor and $\pi'_{\mathcal{Y}}$ projects onto the last factor. Our assumption of *complete visibility* implies that, when $m \geqslant 2$, the branched covers $\pi_{\mathcal{Y}}$ and $\pi_{\mathcal{C},\mathcal{Y}}$ are birationally equivalent. Indeed, letting $Y = (x, \ell)$ consist of points $x = (x_{1,1}, \ldots, x_{m,p})$ and lines $\ell = (\ell_{1,1}, \ldots, \ell_{m,l})$ in the m views. Each point $x_{v,i} \in \mathbb{P}^2$ in a view v is pulled back via the v-th camera to a line in 3-space. As $m \geqslant 2$, the m pull-back lines for generic

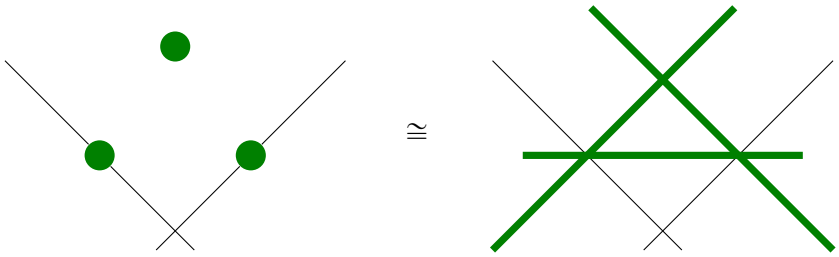


Figure 3.3: Transforming a point-line arrangement to an arrangement of visible lines.

$x_{1,i}, \ldots, x_{m,i}$ intersect in a unique point in $\mathbb{P}^3$. Similarly, each line $\ell_{v,j}$ in a view v is pulled back via the v-th camera to a plane in $\mathbb{P}^3$. As $m \geqslant 2$, the m generic pull-back planes for $\ell_{1,j}, \ldots, \ell_{m,j}$ intersect in a unique line in $\mathbb{P}^3$.

Computationally, working with the branched cover $\pi_{\mathcal{Y}}$ is easier than $\pi_{\mathcal{C},\mathcal{Y}}$ or $\Phi_{p,l,\mathcal{I},m}$, since there are fewer variables. To describe implicit equations vanishing on Inc', it is also convenient to reparametrize $\mathcal{Y}$ solely in terms of lines.

We consider two types of constraints. The first type of constraint is a *line correspondence constraint*: if $\ell_1, \ldots, \ell_m$ are images of the same world line, with respective homogeneous coordinates $\mathbf{l}_1, \ldots, \mathbf{l}_m \in \mathbb{C}^{3\times 1}$, then

$$\mathrm{rk}\begin{bmatrix} P_1^T\mathbf{l}_1 & P_2^T\mathbf{l}_2 & \ldots & P_m^T\mathbf{l}_m \end{bmatrix} \leqslant 2. \tag{3.10}$$

That is, the planes with homogeneous coordinates $P_i^T\mathbf{l}_i$ share a common line in $\mathbb{P}^3$. We distinguish two classes of lines in $\mathbb{P}^2$:

(1) *Visible lines* define valid line correspondences. Besides $m\,l$ observed lines in the joint image, for generic x there is a unique visible line between any two observed points. Taken across all views, any pair of points thus provides a line correspondence which must be satisfied. This scheme is illustrated for the Chicago problem in Figure Figure 3.3

(2) Two generic visible lines suffice to define a point. We may use an additional set of non-corresponding *ghost lines* to define any points which are incident to fewer than two visible lines. A generic ghost line contains exactly one observed point, and is not constrained by a line correspondence constraint. It is simply a device for generating equations.

Thus, from both visible and ghost lines, we obtain *common point* constraints: given

visible and ghost lines $\mathbf{l}_{v,1}, \dots \mathbf{l}_{v,k_i}$ which meet $x_{v,i}$, the projection of the i-th point in the view $v \in 1, \dots, m$, we must have

$$\operatorname{rk}\begin{bmatrix} P_1^T \mathbf{l}_{1,1} & \dots & P_m^T \mathbf{l}_{m,k_i} \end{bmatrix} \leqslant 3, \quad i = 1, \dots, p. \tag{3.11}$$

Assuming that $(p, l, \mathcal{I}, m)$ is a minimal problem, line correspondences and common point constraints give a well-constrained system for the branched cover $\pi'_{\mathcal{Y}} : \mathrm{Inc}' \to \mathcal{Y}_{p,l,\mathcal{I},m}$. Moreover, it is easy to sample Inc' by first sampling Inc (fabricating a scene and cameras) and then projecting. Thus, the two essential ingredients for running Algorithm 2 are in place. Surprisingly, these equations may cut out a variety with multiple components besides the geometrically relevant Inc' in certain cases. We discovered this phenomenon for the 2-camera minimal problems  ,  (both degenerate cases of the five-point problem) and the 3-camera problem  . Computing the number of solutions to these equations for random data using Gröbner bases resulted in a higher degree than achieved by monodromy. The explanation is that each of these problems has "parasitic solutions" which satisfy Equation 3.10 and Equation 3.11, but are not points in the fibers of $\pi'_{\mathcal{Y}}$. For instance, for the problem these equations generically have 80 solutions in camera matrices—however, 16 of these solutions are such that the three recovered camera rays mapping to an observed point with four incidences coincide—in other words, the solution reconstructed from three point correspondences is actually a line! Likewise, there are 4 and 8 excess solutions in camera matrices, respectively, for the problems  ,  . When lifted to the world, the resulting points are not in the desired configuration. In symbolic Gröbner basis computation, these geometric anomalies can be removed by saturating out an appropriate ideal of non-maximal minors from the rank-constrained common line and common point matrices. When using monodromy, they are naturally avoided.

3.2.3 Monodromy of point-line problems

In this section, we provide details on how the numerical computations are set up. Each of the degree computations for $m = 2, 3, 4, 5$ cameras all terminated in a matter of hours or less, using `MonodromySolver` on a laptop machine with 16 GB RAM and a maximum

clock rate of 3.5 GHz. The minimum criterion used for a successful monodromy run was *node consensus*—if the number of solutions known at two different nodes is different, we declare that the monodromy run failed and have to start over. For $m = 2, 3$, the degrees obtained by monodromy were validated using Gröbner bases. For $m = 4, 5$, the degrees we report were obtained consistently from several runs with different random seeds, using a graph with two nodes and four edges and the endpoint randomization scheme described below. For $m = 6$, the computation ran for days, encountering memory issues and failing to result in node consensus. For the problems of higher degree, we found it necessary to use more conservative path-tracking tolerances (eg. `MaxCorrectorSteps => 2` instead of the default 3, `tStepMin => 10^-8` instead of the default 10^{-6}.)

Each of the rotation matrices $\mathbf{R}_2, \ldots, \mathbf{R}_m$ was represented using the rational quaternion parametrization of $\mathbb{P}^3 \dashrightarrow \mathrm{SO}_{\mathbb{C}}(3)$. Letting $[w : x : y : z :]$ be homogeneous coordinates on $\mathbb{P}^3$, we have

$$\mathbf{R} = (w\mathbf{I} - [(x, y, z)]_\times)(w\mathbf{I} + -[(x, y, z)]_\times)^{-1}.$$

Together with the translation vectors $\mathbf{t}_2, \ldots, \mathbf{t}_m$, we construct for each point-line problem a polynomial system of equations in $7\,(m-1)$ unknowns. Among these equations, $m-1$ are affine charts on the domain of the quaternion map, whose coefficients are 5 independent parameters, and an m-th chart for $\mathbf{t}_2$ with 4 independent parameters is used, for a total of $5m-1$ chart parameters. The remaining equations are the line correspondence and common point constraints obtained by taking suitable minors in Equation 3.10 and Equation 3.11. To parametrize the lines in each image, it is natural to use homogeneous coordinates—ie), represent each line $\mathbf{l} \in \mathbb{C}^{3\times 1}$ with three independent parameters. This over-parametrization is harmless for problems where at most two lines are incident at any point. However, cases where three or more lines intersect at some point must be handled with more care: in such cases, we may use three independent parameters for two of the lines, say $\mathbf{l}_1,\ \mathbf{l}_2 \in \mathbb{C}^{3\times 1}$, and the parametrize the remaining lines to have the form $a_1\mathbf{l}_1 + a_2\mathbf{l}_2$, with a_1, a_2 independent parameters. To distinguish lines according to this scheme, we say $\mathbf{l}_1, \mathbf{l}_2$ are independent lines and $a_1\mathbf{l}_1 + a_2\mathbf{l}_2$ is a dependent line. To randomize multiple edges between the same nodes in the homotopy graph, we generate random scalars $\gamma \in \mathbb{C}$ of modulus 1 for each

of the chart equations and each of the independent lines, and multiply the corresponding parameters by each γ. Having randomized independent lines $\mathbf{l}_1 \mapsto \gamma_1 \mathbf{l}_1$, $\mathbf{l}_2 \mapsto \gamma_2 \mathbf{l}_2$, we can maintain the common point and line constraints by sending $(a_1,\, a_2) \mapsto (\gamma_1^{-1} a_1, \gamma_2^{-1} a_2)$ for each of the remaining dependent lines.

We peform a short computational experiment to illustrate how this endpoint randomization scheme can boost the success probability of the degree computation. In this experiment, we use a homotopy graph which is a complete graph on four vertices, with two edges between two vertices. We consider the minimal problem  , whose Galois/monodromy group $\mathrm{Mon}(X/Z) \hookrightarrow S_{32}$ has order 512, is highly imprimitive, and cannot be generated by fewer than 5 permutations (see Result 2.) On the other hand, the probability of generating a transitive subgroup of $\mathrm{Mon}(X/Z)$ with three generators drawn uniformly roughly equals $.52$, as reveleated by simulating one million trials in GAP. Without randomizing endpoints, the multiple edges are redundant, and despite the Betti number of the homotopy graph being 9 we can only ever compute a subgroup of $\mathrm{Mon}(X/Z)$ generated by 3 elements. We find that $4/10$ trials without randomization terminate with the correct number of solutions, which is 32. By contrast, $10/10$ trials, each taking a few minutes, terminate with node consensus using the edge randomization scheme described above. We also note that under a uniform model, the probability of generating $\mathrm{Mon}(X/Z)$ with 9 random elements is roughly $.94$.

3.3 Point-line minimal problems with partial visibility

It is natural to ask what happens when we drop the assumption of complete visibility from the previous section. In three or more views, features such as points and lines might only be partially matched due to occlusions or failures of the matching algorithm that is used. Partial visibility is also a natural setting for considering relaxations of overconstrained minimal problems For instance, Kileel [68] considered a number of relative pose problems which can be understood by taking special linear slices of the so-called calibrated trifocal variety. One of these problems is a natural relaxation of the notorious problem of computing relative orientation from four points in three views [90]. In this relaxation, we impose only three

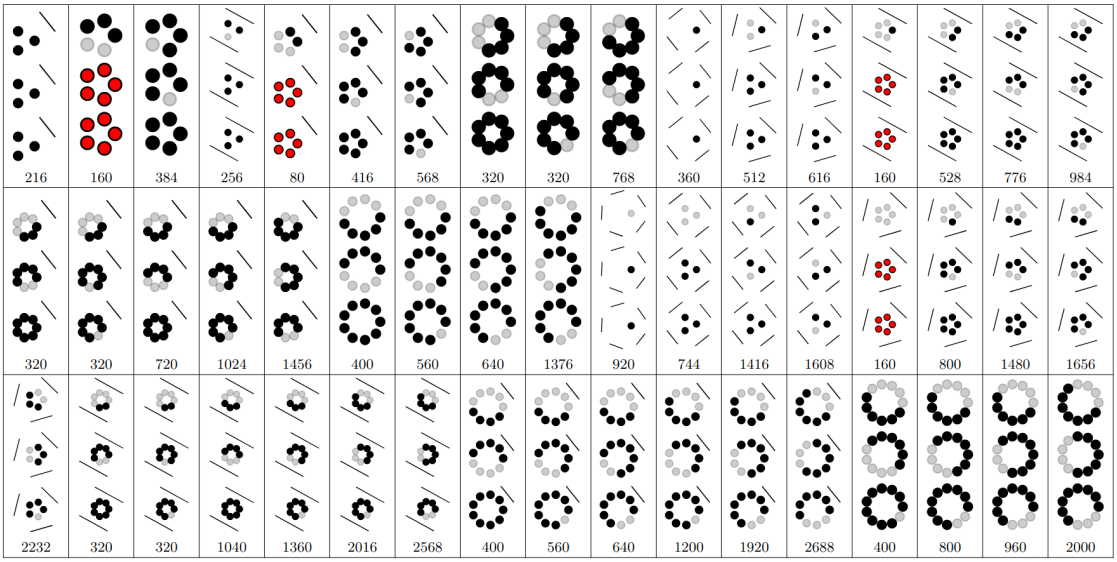
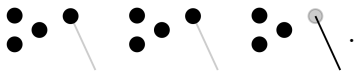


Figure 3.4: Minimal problems with missing three calibrated views with partial visbility and no incidences, together with their degrees. Five-point subproblems indicated in red.

point-point-point correspondences and one point-point-line correspondence. Pictorially:

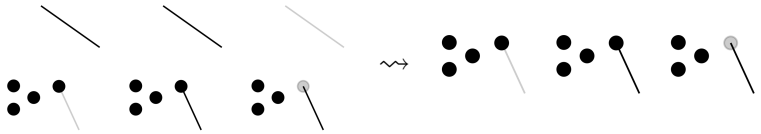


This problem is not minimal in the sense of the previous section—we cannot recover an arbitrary world line that passes through the fourth point. However, counting dimensions and computing Jacobians does show that this problem does have finitely many solutions in cameras for generic data—272 to be exact. Thus, in the setting of partial visibility, we distinguish between minimal and *camera-minimal* problems. Moreover, in a setting where we actually have four complete point correspondences, we could easily obtain a complete line correspodence. This leads us to the following problem, which is both minimal and camera-minimal:

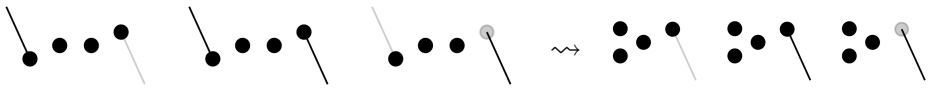
The joint camera maps for the above problems are not birationally equivalent, but they are inter-reducible in a sense that can be made precise [35].

We use a similar encoding as the previous section for point-line problems with partial visibility—$(p, l, \mathcal{I}, \mathcal{O})$, where now $\mathcal{O} = ((\mathcal{P}_1, \mathcal{L}_1), \ldots, (\mathcal{P}_m, \mathcal{L}_m))$ is a set of *observations*

in each view with $\mathcal{P}_i, \mathcal{L}_i \subset [l]$. With the same assumptions on $\mathcal{I}$ as before, we also assume that if $(x, \ell_1),\ (x, \ell_2) \in \mathcal{I}$ and $\ell_1, \ell_2 \in \mathcal{L}_i$, then also $x \in \mathcal{P}_i$. We once again define the joint camera map $\mathcal{X}_{p,l,\mathcal{I}} \times \mathcal{C}_m \dashrightarrow \mathcal{Y}_{p,l,\mathcal{I},\mathcal{O}}$ and ask for which $(p, l, \mathcal{I}, \mathcal{O})$ it is a branched cover. The first interesting case is for $m = 3$ cameras. Like in the previous section, there are infinitely many minimal problems in this case, since we could always, say, add an arbitrary number of line-line-nothing correspondences to a given minimal problem. Thus we have a reduction



A number of analagous reductions arise from partial visibility of incidences. For instance,



Formally, a reduction between two *minimal* problems $(p, l, \mathcal{I}, \mathcal{O}) \rightsquigarrow (p', l', \mathcal{I}', \mathcal{O}')$ has the following properties (cf. [35, Definition 5]):

1) for all $x \in [p] \backslash [p']$, we have $\#\{(x, \ell) \in \mathcal{I}'\} \leqslant 1$

2) the diagram below commutes such that the generic fiber in $\mathcal{X}_{p,l,\mathcal{I},m}$ maps bijectively onto the corresponding fiber in $\mathcal{X}_{p',l',\mathcal{I}',\mathcal{O}'}$:

With this notion of reduction, we can prove that there are finitely many reduced minimal problems in which each line is incident to at most one point—see [35, Theorems 2 & 7]. We can count their number by counting dimensions, although it is difficult to do so by hand—there are $140,616$ problems up to relabeling of cameras. Moreover, these problems divide into $74,575$ equivalence classes under a "swap & label" operation, such that the maps $\mathrm{Inc}' \dashrightarrow \mathcal{Y}_{p,l,\mathcal{I},\mathcal{O}}$ in each class are all birationally equivalent [35, see Corollary 1 and discussion]. Among these problems, we were able to verify the degrees of 66 problems first appearing in [68]. For problems without incidences, the situation is much more simple. There are 51 reduced minimal problems, given in Figure 3.4, none of which are swap &

Table 3.2: Distribution of degrees < 300 of minimal problems for three calibrated views with partial visbility.

camera-degree	64	80	144	160	216	224	240	256	264	272	288
# problems	13	9	3	547	7	2	159	2	2	11	4

label equivalent.

To search for problems of low degree, multiple instances of `MonodromySolver` were run on problems representative of these equivalence classes on a computing cluster at the Czech Institute of Informatics, Robotics, and Cybernetics, truncating each computation after 300 or more solutions were found. A visible/ghost line parametrization similar to that of the previous section was used. The minimal problems of degre < 300 are given in Table 3.2. The high frequency of certain degrees (eg. 160) leads us to speculate that many of these problems are birationally equivalent. We also expect that many of these low-degree problems have imprimitive Galois/monodromy groups—this will be the main topic of the next chapter.

CHAPTER 4
DECOMPOSITION

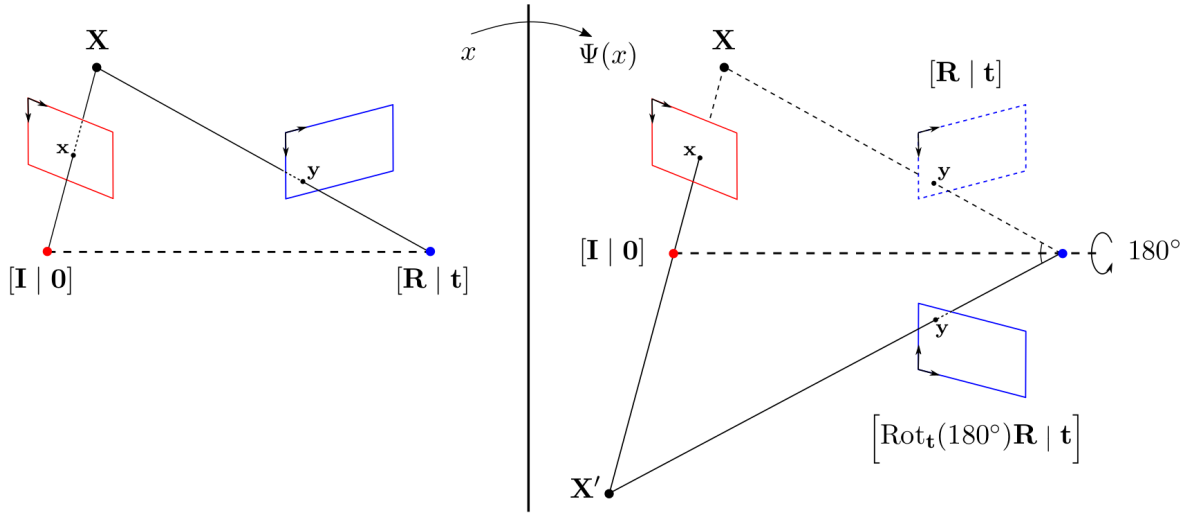


Figure 4.1: The twisted pair symmetry.

We consider once again the five-point problem from section 3.1. This is ⚄ from Table 3.1. Although the degree of the joint camera map is 20, solutions to this problem map $2-1$ to 10 essential matrix solutions. Solutions mapping to the same essential matrix are related by the twisted pair symmetry, illustrated above. The goal of this chapter is to search for analogous decompositions for other minimal problems, outlining some of the general theory along the way.

For the five-point problem, we are interested in the following system of polynomial equations and inequations:

$$\begin{aligned} \mathbf{R}^\top\mathbf{R} = \mathtt{I}, \quad \det\mathbf{R} = 1, \\ \beta_i\mathbf{y}_i = \mathbf{R}\alpha_i\mathbf{x}_i + \mathbf{t}, \quad \alpha_i,\beta_i \neq 0, \quad \forall\, i = 1,\ldots,5. \end{aligned} \tag{4.1}$$

An inherent ambiguity of the five-point problem is that the unknowns $\mathbf{t}, \alpha_1, \ldots, \alpha_5, \beta_1, \ldots, \beta_5$ can only be recovered up to a common scale factor. If we treat these unknowns as homogeneous coordinates on a 12-dimensional projective space, then we are led to consider the branched cover $f : X \to Z$, where X is the incidence correspondence

$$X = \{(\mathbf{R}, (\mathbf{t}, \alpha_1, \ldots, \beta_5), (\mathbf{x}_1, \ldots, \mathbf{y}_5)) \in \mathrm{SO}_{\mathbb{C}}(3) \times \mathbb{P}^{12}_{\mathbb{C}} \times Z \mid \text{ Equation 4.1 holds } \}.$$

For most solutions to Equation 4.1, the associated *twisted pair* solution depicted in Figure 4.1 is obtained by rotation of the second camera frame 180° about the baseline connecting the first and second camera centers. Algebraically, the twisted pair may be viewed as a rational map $\Psi : X \dashrightarrow X$, given coordinate-wise by

$$\begin{aligned}
\Psi(\mathbf{R}) &= \left(2\frac{\mathbf{t}\mathbf{t}^\top}{\mathbf{t}^\top\mathbf{t}} - \mathbf{I}\right)\mathbf{R} \\
\Psi(\mathbf{t}) &= \mathbf{t} \\
\Psi(\alpha_i) &= \frac{-\alpha_i\|\mathbf{t}\|^2}{\|\mathbf{t}\|^2 + 2\langle \mathbf{R}^\top\mathbf{t}, \alpha_i\mathbf{x}_i\rangle} = \frac{-\alpha_i\|\mathbf{t}\|^2}{\|\beta_i\mathbf{y}_i\|^2 - \|\alpha_i\mathbf{x}_i\|^2} \\
\Psi(\beta_i) &= \frac{\beta_i\|\mathbf{t}\|^2}{\|\mathbf{t}\|^2 + 2\langle \mathbf{R}^\top\mathbf{t}, \beta_i\mathbf{x}_i\rangle} = \frac{\beta_i\|\mathbf{t}\|^2}{\|\beta_i\mathbf{y}_i\|^2 - \|\alpha_i\mathbf{x}_i\|^2} \\
\Psi(\mathbf{x}_1, \ldots, \mathbf{x}_5, \mathbf{y}_1, \ldots, \mathbf{y}_5) &= (\mathbf{x}_1, \ldots, \mathbf{x}_5, \mathbf{y}_1, \ldots, \mathbf{y}_5).
\end{aligned} \tag{4.2}$$

Here, we use the notation $\langle , \rangle$ and $\|\cdot\|^2$ for the complex quadratic forms $\langle \boldsymbol{a}, \boldsymbol{b}\rangle = a_1b_1 + a_2b_2 + a_3b_3$, $\|\boldsymbol{a}\|^2 = \langle \boldsymbol{a}, \boldsymbol{a}\rangle$ which restrict to the usual norm and inner product on $\mathbb{R}^3$. We note that Ψ is undefined whenever $\boldsymbol{t} \in \mathbb{P}^2$ is an *isotropic vector* satisfying $\|\boldsymbol{t}\|^2 = 0$, and whenever $\|\alpha_i\mathbf{x}_i\|^2 = \|\beta_i\mathbf{y}_i\|^2$ for some $i = 1, \ldots, 5$. The second condition can be understood geometrically: if the camera centers and the world point $\mathbf{X} = \alpha\mathbf{x}$ form an isosceles triangle with base $\|\mathbf{t}\|$, then, after rotating the second camera, the rays which join camera centers to the respective image points will become parallel.

One can check (e.g. [85, p. 20]) that we have an equality of mappings $f \circ \Psi = f$ wherever Ψ is defined. The map Ψ is a deck transformation (see Definition 4.1.3) of the

branched cover f. To remove the twisted pair symmetry, we recall the *essential variety*

$$\mathcal{E} = \{\mathbf{E} \in \mathbb{P}(\mathbb{C}^{3\times 3}) \mid \det \mathbf{E} = 0,\ \mathbf{E}\mathbf{E}^\top\mathbf{E} - \frac{1}{2}\operatorname{tr}(\mathbf{E}\mathbf{E}^\top)\mathbf{E} = 0\} \tag{4.3}$$

and define

$$Y = \{(\mathbf{E},\, (\mathbf{x}_1,\, \ldots,\, \mathbf{y}_5)) \in \mathcal{E} \times Z \mid \mathbf{y}_i^\top \mathbf{E}\, \mathbf{x}_i = 0,\, i = 1,\, \ldots,\, 5\}.$$

This gives rise to a *factorization* of the branched cover $X \dashrightarrow Z$, which is simply a commutative diagram

$$\begin{array}{ccc} X & \dashrightarrow & Y \\ & \searrow & \downarrow \\ & & Z \end{array} \tag{4.4}$$

such that $X \dashrightarrow Y$ and $Y \dashrightarrow Z$ are branched covers. The map $X \dashrightarrow Y$ is given by

$$(\mathbf{R}, \mathbf{t},\, (\mathbf{x}_1,\, \ldots,\, \mathbf{y}_5)) \mapsto ([\mathbf{t}]_\times \mathbf{R},\, (\mathbf{x}_1,\, \ldots,\, \mathbf{y}_5))\,. \tag{4.5}$$

We have that $\operatorname{Mon}(X/Z) \cong S_2 \wr S_{10} \cap A_{20}$.

State-of-the art five-point solvers such as Nistér's [89] are based on the formulation in terms of essential matrices. From the ten essential matrix solutions corresponding to a fiber of the map $Y \dashrightarrow Z$, the rotation/translation pairs corresponding to any *real* essential matrix may be computed via SVD [53, Sec 9.6], which in turn allow for easily recovering the unknown depths.

It is natural to consider whether analogues of the twisted pair or essential matrix exist for the problems of section 3.2 and section 3.3. Propositions 4.1.4 and 4.1.1 tell us that the existence of nontrivial deck transformations and decomposability can be decided just by looking at generators of the Galois/monodromy group. These facts follow from standard results—from Galois theory or topology, depending on the point of view taken. However, they seem to be under-appreciated in the worlds of engineering and applied math. In this chapter, we demonstrate that numerically computing Galois/monodromy groups gives a powerful tool for detecting hidden symmetries in equations coming from applications.

4.1 Preliminary notions

Consider a factorization as in Equation 4.4. If $\deg(X/Y)$ and $\deg(Y/Z)$ are both strictly less than $\deg(X/Z)$, we say that the factorization is *proper* and that the branched cover $X \dashrightarrow Z$ is *decomposable*. Otherwise, $X \dashrightarrow Z$ is *indecomposable.*

Proposition 4.1.1 below implies that the decomposability of a branched cover can be determined from the Galois/monodromy group alone. We recall that a *block system* for the monodromy action $\mathrm{Mon}(X/Z) \curvearrowright X_z = \{x_1 \dots x_d\}$, is a partition of $X_z = B_1 \cup \cdots \cup B_k$, comprised of equally-sized blocks $B_1, \dots, B_k$, which is preserved in the sense that blocks are always mapped to blocks under the group action. The block systems associated to the action form a lattice under refinement, whose respective maximum and minimum elements are $\{X_z\}$ and $\{\{x_1\}, \dots, \{x_d\}\}$. If any other block systems exist, then $\mathrm{Mon}(X/Z)$ is said to be *imprimitive*, and otherwise it is *primitive*.

Example 7. Consider the degree-6 branched cover over $Z = \mathbb{C}^2$ given by

$$X = \{(x_1, x_2, x_3, a, b) \in \mathbb{C}^3 \times \mathbb{C}^2 \mid x_1 + x_2 + x_3 = x_2^2 + x_2x_3 + x_3^2 + a = x_3^3 + ax_3 + b = 0\}.$$

$X \to Z$ factors as a composition of degree-2 and degree-3 branched covers: we may take

$$Y = \{(x_3, a, b) \in \mathbb{C}^1 \times \mathbb{C}^2 \mid x_3^3 + ax_3 + b = 0\}.$$

In this example we have $\mathbb{C}(X) \cong \mathbb{C}(Y)^{\mathrm{gal}}/\mathbb{C}(Z)$. The group $\mathrm{Mon}(X/Z)$ is given, up to conjugacy, by the left-regular representation of $S_3 \hookrightarrow S_6$. This holds more generally for Galois covers due to the normal basis theorem [7, Theorem 28].

Given a factorization Equation 4.4, we have $\deg(X/Z) = \deg(X/Y)\ \deg(Y/Z)$, and a partition

$$X_z = X_{y_1} \cup \cdots \cup X_{y_k} \tag{4.6}$$

with $k = \deg(Y/Z)$. The proof of Proposition 4.1.2 below shows that this is a block system for the monodromy action with blocks of size $\deg(X/Y)$. Conversely, imprimitivity implies decomposability.

Proposition 4.1.1. A branched cover is decomposable if and only if $\mathrm{Mon}(X/Z)$ is imprimitive.

Proposition 4.1.1 dates back to the work of Ritt [95], who characterized the possible decompositions of branched covers $\mathbb{C} \ni x \mapsto p(x) \in \mathbb{C}$ given by a univariate polynomial p. A Galois-theoretic proof of Proposition 4.1.1 may be found, for instance, in [24]. If we know $\mathrm{Mon}(X/Z)$, it is also possible to identify $\mathrm{Mon}(X/Y)$ and $\mathrm{Mon}(Y/Z)$ occuring in the factorization of Equation 4.4, as the next proposition shows.

Proposition 4.1.2. Consider a factorization of branched covers as in Equation 4.4. For fixed generic $z \in Z$, partition X_z as in Equation 4.6. The action $\mathrm{Mon}(X/Z) \curvearrowright X_z$ induces two other group actions which are equivalent to the monodromy groups of the individual factors:

1) action on blocks: $\mathrm{Mon}(X/Z) \curvearrowright \{X_{y_1}, \ldots, X_{y_k}\}$, which is equivalent to $\mathrm{Mon}(Y/Z)$.

2) action on a single block: $\mathrm{Mon}(X/Z)_{X_y} \curvearrowright X_y$, where $\mathrm{Mon}(X/Z)_{X_y}$ denotes the stabilizer of the set X_y under the action by $\mathrm{Mon}(X/Z) \curvearrowright X_z$. This is equivalent to $\mathrm{Mon}(X/Y)$, and thus independent of the choice $y \in Y_z$.

Proof. 1) For each $\sigma_\gamma \in \mathrm{Mon}(X/Z)$, there is an induced permutation of the blocks:

$$\widetilde{\sigma_\gamma} = \begin{pmatrix} X_{y_1} & \cdots & X_{y_k} \\ \sigma_\gamma(X_{y_1}) & \cdots & \sigma_\gamma(X_{y_k}) \end{pmatrix}.$$

Indeed, suppose that $x, x' \in X_{y_i}$ are such that $\sigma_\gamma(x) \in X_{y_j}, \sigma_\gamma(x') \in X_{y_k}$, and consider the lift $\widetilde{\gamma} : [0,1] \to Y$ starting at y_i. We must have both $\widetilde{\gamma}(1) = y_j$ and $\widetilde{\gamma}(1) = y_k$. Hence $k = j$ by the unique path-lifting property applied to $Y \dashrightarrow Z$, showing that σ_γ preserves the partition into blocks. In this way we get a group homomorphism

$$\begin{aligned} \mathrm{Mon}(X/Z) &\to \mathit{Sym}(\{X_{y_1}, \ldots, X_{y_k}\}) \\ \sigma_\gamma &\mapsto \widetilde{\sigma_\gamma}, \end{aligned} \tag{4.7}$$

which represents the action of $\mathrm{Mon}(X/Z)$ on the blocks. Now, there is also an injective

group homomorphism

$$\begin{aligned} \mathrm{Mon}(Y/Z) &\to Sym(\{X_{y_1}, \ldots, X_{y_k}\}) \\ \tau_\gamma &\mapsto \begin{pmatrix} X_{y_1} & \cdots & X_{y_k} \\ X_{\tau_\gamma(y_1)} & \cdots & X_{\tau_\gamma(y_k)} \end{pmatrix} \end{aligned} \tag{4.8}$$

obtained by restricting the natural isomorphism $Sym(Y_z) \cong Sym(\{X_{y_1}, \ldots, X_{y_k}\})$ that identifies a point $y_i \in Y_z$ with its corresponding block X_{y_i}. We wish to show that maps in Equation 4.7 and Equation 4.8 have the same image. This follows easily if we restrict γ in both maps to be loops contained in a regular locus for $X \dashrightarrow Z$: the lifts of γ to Y (which, by our restriction, also lift to X) determine the corresponding permutation of blocks in X, and vice-versa. Indeed, we have $\sigma_\gamma(X_y) = X_{\tau_\gamma(y)}$ for any $y \in Y_z$. To see this, it is enough to show one set is contained in the other. A point $x \in \sigma_\gamma(X_y)$ is the endpoint of some lift of γ to X. The image of this lift in Y is itself a lift $\widetilde{\gamma} : [0,1] \to Y$ with $\widetilde{\gamma}(0) = y$—hence $\widetilde{\gamma}(1) = \tau_\gamma(y)$, and the endpoint of our original lift x is in $X_{\tau_\gamma(y)}$.

2) The proof amounts to showing that a loop γ in Z lifts to a loop in Y if and only if $\sigma_\gamma \in \mathrm{Mon}(X/Z)$ stabilizes each of the blocks. As in the previous part, this only true if we consider loops in a suitably small regular locus $U \subset Z$. It suffices to take U contained in a regular locus for $X \dashrightarrow Z$ and whose preimage in Y is a regular locus for $X \dashrightarrow Y$. □

Proposition 4.1.1 shows that an arbitrary branched cover $X \dashrightarrow Z$ factors as a composition of indecomposable branched covers:

$$X = Y_0 \dashrightarrow Y_1 \dashrightarrow \cdots \dashrightarrow Y_{k-1} \dashrightarrow Y_k = Z. \tag{4.9}$$

Such a factorization corresponds to a maximal chain in the lattice of block systems, and the associated degrees can be read off from the block sizes. Equivalently, for any $x \in X_z$, a maximal chain in the lattice of blocks of $\mathrm{Mon}(X/Z)$ corresponds to a chain of subgroups that contain the stabilizer $\mathrm{Mon}(X/Z)_x$ (cf. [96, proof of Theorem 9.15])

$$\mathrm{Mon}(X/Z)_x = G_0 \subset G_1 \subset \cdots \subset G_{k-1} \subset G_k = \mathrm{Mon}(X/Z),$$

and we have $\deg(Y_i/Y_{i+1}) = [G_{i+1} : G_i]$ for $i = 0, \ldots, k-1$. The decomposition of Equation 4.9 is not unique. In fact, as Ritt already understood [95, p. 53], there are many examples where even the multi-set of degrees $\deg(Y_i/Y_{i+1})$ is not unique.

Example 8. This example was first given in [50, Example 25]. Consider

$$X = \left\{(x,z) \in \mathbb{C}^1 \times \mathbb{C}^1 \mid \left(x \cdot (x+6) \cdot \left(x^2 - 6x + 36\right)\right)^3 = z\,\left((x-3)\left(x^2+3x+9\right)\right)^3\right\}.$$

This is a Galois cover of degree 12 over $Z = \mathbb{C}^1$. The lattice of block systems can be identified with the lattice of subgroups of the Galois/monodromy group A_4. Corresponding to a subgroup chain

$$id \subset C_2 \subset V_4 \subset A_4$$

is the decomposition of $X \to Z$ into rational branched covers of degrees $(3,2,2)$ implied by the identity

$$\frac{\left(x \cdot (x+6) \cdot (x^2-6x+36)\right)^3}{\left((x-3)\left(x^2+3x+9\right)\right)^3} = x^3 \circ \frac{x(x-12)}{x-3} \circ \frac{x(x+6)}{x-3}.$$

Likewise, the chain

$$id \subset C_3 \subset A_4$$

corresponding to

$$\frac{\left(x\ (x+6) \cdot (x^2-6x+36)\right)^3}{\left((x-3)\left(x^2+3x+9\right)\right)^3} = \frac{x^3(x+24)}{x-3} \circ \frac{x(x^2-6x+36)}{x^2+3x+9}.$$

gives a decomposition with degrees $(4,3)$.

Finally, we define and carefully study the deck transformations of a branched cover, of which the twisted pair symmetry from the introduction is a special case.

Definition 4.1.3. A birational equivalence from a branched cover to itself which fixes the base is called a *deck transformation.* Explicitly, for $f : X \dashrightarrow Z$ a deck transformation $\Psi : X \dashrightarrow X$ must satisfy $f \circ \Psi = f$ whenever both maps are defined. The deck transformations form a group under composition which acts on a generic fiber X_z. The deck

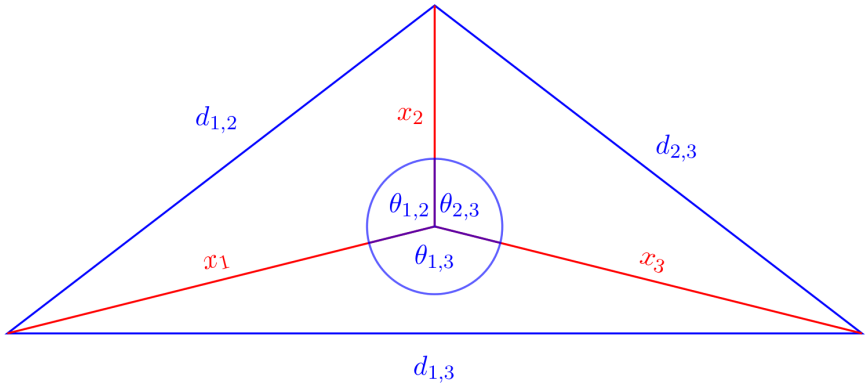


Figure 4.2: Frontal view of the P3P problem: x_1, x_2, x_3 are unknown.

transformation group can be naturally identified with the automorphisms of $\mathbb{C}(X)$ which fix $\mathbb{C}(Z)$, denoted $\mathrm{Aut}(X/Z)$.

Analogously to decomposability, Proposition 4.1.4 shows that the existence of a non-trivial deck transformation can be decided from the Galois/monodromy group alone. This turns out to be stronger than decomposability in general. We learned of Proposition 4.1.4 from the sources [9, 28]. Since it seems less well-known outside of the literature on Galois/monodromy groups, we give a self-contained proof. In topology, a deck transformation of a covering map f can be any continuous function Ψ satisfying $f \circ \Psi = f$. Our proof of Proposition 4.1.4 reveals that, for a rational branched cover f with regular locus U, the deck transformations of $f_{|f^{-1}(U)}$ in the topological sense are always rational maps in the sense of Definition 4.1.3. Before giving the proof, we first consider three illustrative examples.

Example 9. Let $X = \mathbf{V}(x^2 + ax + b) \subset \mathbb{C}^3$, $Z = \mathbb{C}^2$ and $f : X \to Z$ be the degree-2 branched cover defined by coordinate projection $f(x, a, b) = (a, b)$. The deck transformation defined by $\Psi(x, a, b) = (-x - a, a, b)$ acts on a generic fiber $X_{(a,b)}$ by permuting the two roots of the quadratic equation $x^2 + ax + b = 0$.

Example 10. Ask et al. [8] define a polynomial system $F(\boldsymbol{x})$ with p-fold symmetry to be such that $F(\boldsymbol{x}) = 0$ implies $F(\omega\, \boldsymbol{x}) = 0$ whenever ω is a p-th root of unity. For example, the equations

$$
\begin{aligned}
f_{1,2} &= x_1^2 + x_2^2 - c_{1,2}x_1x_2 - d_{1,2}^2 \\
f_{1,3} &= x_1^2 + x_3^2 - c_{1,3}x_1x_3 - d_{1,3}^2 \\
f_{2,3} &= x_2^2 + x_3^2 - c_{2,3}x_2x_3 - d_{2,3}^2
\end{aligned}
\tag{4.10}
$$

have a 2-fold sign symmetry: $(x_1, x_2, x_3) \mapsto (-x_1, -x_2, -x_3)$. These equations define the famous Perspective-3-Point problem or *P3P problem*. Here, each $c_{i,j}$ is equal to $2\cos\theta_{i,j}$ as in Figure 4.2. Letting X denote the vanishing locus of Equation 4.10 in $\mathbb{C}^9$, the coordinate projection onto the space of knowns $\mathbb{C}^6$ is a branched cover with a deck transformation given by the sign-symmetry.

Ask et al. [8] develop algorithms for detecting and exploiting *partial* p-fold symmetries (occurring in only some subset of the variables) in the automatic generation of polynomial solvers. These methods were generalized by Larsson and Åström [74] to the case of *weighted* partial-p fold symmetries. In general, a branched cover with a weighted partial-p fold symmetry will have a deck transformation of order p, degree $e\,p$ for some integer e, and its Galois/monodromy group will be a subgroup of $C_p \wr S_e$.

Example 11. Two models of the same birational equivalence class will have isomorphic deck transformation groups, but the formulas defining the deck transformations may look quite different. We consider an example from [81, 105], in which the author(s) construct moduli spaces obtained by letting the *absolute conic* [53] degenerate to a double line. A particular double cover of the essential variety plays a role in this work. Explicitly, the branched cover $X \to \mathcal{E}$ is given by

$$
X = \{([a_0 : a_1 : a_2 : a_3], [b_0 : b_1 : b_2 : b_3]) \in \mathbb{P}^3 \times \mathbb{P}^3 \mid a_0b_0 + a_1b_1 + a_2b_2 + a_3b_3 = 0\}
$$

$$
([\mathbf{a}], [\mathbf{b}]) \mapsto \begin{pmatrix} a_0b_0 - a_1b_1 - a_2b_2 + a_3b_3 & a_1b_0 + a_0b_1 + a_3b_2 + a_2b_3 & a_2b_0 - a_3b_1 + a_0b_2 - a_1b_3 \\ a_1b_0 + a_0b_1 - a_3b_2 - a_2b_3 & -a_0b_0 + a_1b_1 - a_2b_2 + a_3b_3 & a_3b_0 + a_2b_1 + a_1b_2 + a_0b_3 \\ a_2b_0 + a_3b_1 + a_0b_2 + a_1b_3 & -a_3b_0 + a_2b_1 + a_1b_2 - a_0b_3 & -a_0b_0 - a_1b_1 + a_2b_2 + a_3b_3 \end{pmatrix},
$$

and there exists a birational equivalence

$$\begin{array}{ccc} X & \dashrightarrow & \mathrm{SO}_{\mathbb{C}}(3) \times \mathbb{P}^2 \\ \downarrow & & \downarrow \\ \mathcal{E} & \longrightarrow & \mathcal{E}. \end{array}$$

where the bottom map is the identity. The top map may be given by

$$([\boldsymbol{a}], [\boldsymbol{b}]) \mapsto \left((a_3\, I + [\mathbf{a}]_\times)([\mathbf{a}]_\times - a_3\, I)^{-1},\ \begin{pmatrix} 2\,a_1b_0 - 2\,a_0b_1 + 2\,a_3b_2 - 2\,a_2b_3 \\ 2\,a_2b_0 - 2\,a_3b_1 - 2\,a_0b_2 + 2\,a_1b_3 \\ 2\,a_3b_0 + 2\,a_2b_1 - 2\,a_1b_2 - 2\,a_0b_3 \end{pmatrix} \right),$$

where now

$$[\mathbf{a}]_\times = \begin{pmatrix} 0 & a_0 & a_1 \\ -a_0 & 0 & a_2 \\ -a_1 & -a_2 & 0 \end{pmatrix}.$$

For $\mathrm{SO}_{\mathbb{C}}(3) \times \mathbb{P}^2 \dashrightarrow \mathcal{E}$, the action on the fiber applies the twisted pair map as in Equation 4.2. For $X \dashrightarrow \mathcal{E}$, the action swaps coordinates $([\boldsymbol{a}], [\boldsymbol{b}]) \mapsto ([\boldsymbol{b}], [\boldsymbol{a}])$.

Proposition 4.1.4. Let $X \dashrightarrow Z$ be a branched cover and fix generic $z \in Z$. We may identify the deck transformation group with a subgroup of $Sym(X_z)$ by restricting functions to X_z. This permutation group is *equal* to the centralizer of $\mathrm{Mon}(X/Z)$ in $Sym(X_z)$.

Proof. We abbreviate the deck transformation group and centralizer subgroup by D and C, respectively. We define a map between these groups as follows:

$$\begin{aligned} \varphi : D &\to Sym(X_z) \\ \Psi &\mapsto \begin{pmatrix} x_1 & \cdots & x_d \\ \Psi(x_1) & \cdots & \Psi(x_d) \end{pmatrix} \end{aligned}$$

To prove Proposition 4.1.4, we verify the following properties of φ:

1) φ is a group homomorphism.

2) φ is injective.

3) The image of φ is contained in C.

4) C is contained in the image of φ—more explicitly, for all $\sigma \in C$ there exists a deck transformation $\Psi_\sigma \in D$ whose restriction to the fiber X_z equals the permutation σ.

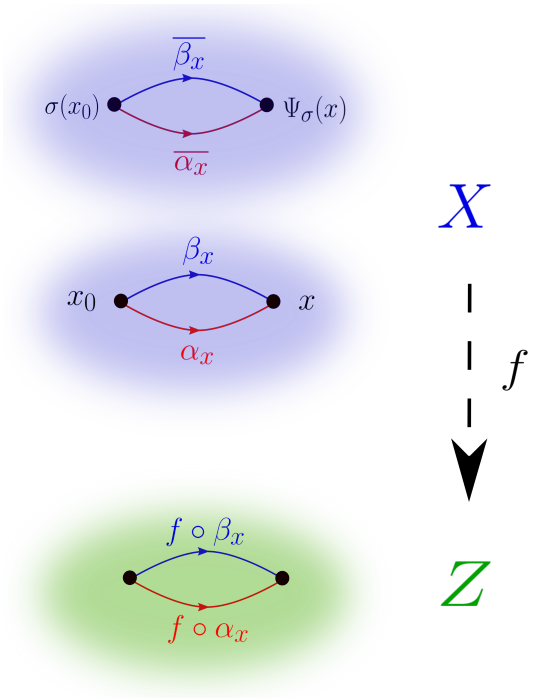


Figure 4.3: Construction and well-definedness of Ψ_σ.

Property 1) is straightforward. Properties 2) and 3) both follow from the unique path-lifting property. For instance, if $\Psi(x_i) = x_i$ for $i = 1, \ldots, d$, then for generic $x \in X$ will be the endpoint of the lift $\widetilde{\gamma}$ of some path γ in Z based at z—if $x_i = \Psi(x_i)$ is the initial point of this lift, then we must have $\Psi \circ \widetilde{\gamma} = \widetilde{\gamma}$, so that in particular $\Psi(x) = \Psi \circ \widetilde{\gamma}(1) = \widetilde{\gamma}(1) = x$. This gives Property 2). The proof of Property 3) is very similar, and may also be found, for instance, in [28, Proposition 1.3].

It remains to show Property 4). We do so by first constructing a map $\Psi_\sigma : X \dashrightarrow X$ pointwise via lifting paths. The argument is analagous to the proof in [54, Propsition 1.39]. Fix $x_0 \in X_z$. For generic $x \in X$, there exists a path $\alpha_x : [0, 1] \to X$ from x_0 to x whose image in Z is contained in a regular locus for $X \dashrightarrow Z$. Define $\overline{\alpha_x}$ to be the lift based at $\sigma(x_0)$ whose image in Z coincides with the image of α. We define

$$\Psi_\sigma : X \dashrightarrow X$$
$$x \mapsto \overline{\alpha_x}(1).$$

First of all, we must show that Ψ_σ is well-defined. This means that for any other path β_x

from x_0 to x we must have $\overline{\beta_x}(1) = \overline{\alpha_x}(1)$. We refer the reader to Figure Figure 4.3 to more easily follow the argument. Consider the loop γ based at $f(x_0)$ in Z obtained by concatenating $\overleftarrow{f \circ \beta_x}$ (the reverse of the path $f \circ \beta_x$) with $f \circ \alpha_x$. Since σ and σ_γ commute, we have that

$$\begin{aligned}\sigma_\gamma(\sigma(x_0)) &= \sigma(\sigma_\gamma(x_0)) \\ &= \sigma(x_0).\end{aligned}$$

Thus σ_γ fixes $\sigma(x_0)$, and it follows that the lift $\widetilde{\gamma}$ based at $\sigma(x_0)$ is a loop. This implies that $\overline{\alpha_x}$ and $\overline{\beta_x}$ have the same terminal point $\Psi_\sigma(x)$, proving well-definedness.

Consider now an arbitrary $x \in X_z$. Note that in this case $f \circ \alpha_x$ is a loop. We calculate

$$\begin{aligned}\sigma(x) &= \sigma(\sigma_{f\circ\alpha_x}(x_0)) \\ &= \sigma_{f\circ\alpha_x}(\sigma(x_0)) \\ &= \Psi_\sigma(x).\end{aligned}$$

Thus, restricting Ψ_σ to X_z yields the permutation σ. Moreover, by definition of Ψ_σ we have that $f \circ \Psi_\sigma = f$ on the locus of points where both maps are defined. It remains to show that Ψ_σ is a rational map, since then it will also follow that $\Psi_{\sigma^{-1}}$ is a rational inverse. First we note that, in a suitably small neighborhood of any generic point $x \in X$, we can write $\Psi_\sigma = g_x \circ f$, where g_x is a holomorphic local inverse of f. Such an expression for Ψ_σ exists for any x in some Zariski open subset of X. It follows that Ψ_σ is a meromorphic map from X to itself—in other words, it is holomorphic after restricting to a Zariski-open $U \subset X$. The fact that Ψ_σ is a rational map now follows from the appropriate "GAGA principle" [98, Theorem 6.1.4]. □

Proposition 4.1.5. A branched cover $X \dashrightarrow Z$ of degree d with a nontrivial deck transformation Ψ is either decomposable or its Galois/monodromy group is cyclic of order d. In the latter case, $\mathrm{Mon}(X/Z)$ is imprimitive precisely when d is composite.

Proof. Partition X_z into the orbits under repeated application of Ψ. This partition is preserved under the monodromy action. Thus, the Galois group is imprimitive if this partition is nontrivial. Otherwise, the action of Ψ on X_z generates a cyclic group $C_d \subset S_d$. Letting $\mathrm{Cent}(\cdot)$ denote the centralizer in S_d, we have $C_d \subset \mathrm{Cent}(\mathrm{Mon}(X/Z))$, which holds if and only if $\mathrm{Mon}(X/Z) \subset \mathrm{Cent}(C_d) = C_d$. Since $\mathrm{Mon}(X/Z)$ is transitive, we must have $\mathrm{Mon}(X/Z) = C_d$. □

In general, a decomposable branched cover need not have any deck transformations. However, a converse to Proposition 4.1.5 does hold in a special case frequently encountered in practice.

Proposition 4.1.6. $X \dashrightarrow Z$ has a deck transformation of order 2 if and only if $\mathrm{Mon}(X/Z)$ has a block of size 2.

Proof. $\Rightarrow$ As in Proposition 4.1.5. $\Leftarrow$ Proposition 4.1.1 gives a factorization such that $\mathbb{C}(X)/\mathbb{C}(Y)$ is a degree 2 extension, and thus always Galois. □

4.2 Decomposing minimal problems

4.2.1 Absolute pose of points and lines

In this section, we apply the mathematical framework of the previous section to absolute pose problems involving combinations of point/line features appearing in work of Ramalingam et al. [93] Although the problems considered here are of low degree, computing the Galois/monodromy groups yields new insights which might be applied to building better solvers for these problems.

We begin formulating these problems in the language of branched covers. Our general task is to determine a calibrated camera matrix $[\mathbf{R} \mid \mathbf{t}]$ from correspondence data between the scene and images. We let p and l be the numbers of point-point and line-line correspondences, respectively, between 3D and 2D. The total space of our branched cover is

$$X_{p,l} = \left(\mathbb{P}^3\right)^p \times \left(\mathbb{G}_{1,3}\right)^l \times \mathrm{SE}_{\mathbb{C}}(3)$$

where $\mathbb{G}_{1,3}$ denotes the Grassmannian of lines in $\mathbb{P}^3$. The base space equals

$$Z_{p,l} = \left(\mathbb{P}^3\right)^p \times \left(\mathbb{G}_{1,3}\right)^l \times \left(\mathbb{P}^2\right)^p \times \left(\mathbb{G}_{1,2}\right)^l,$$

where $\mathbb{G}_{1,2}$ denotes the Grassmannian of lines in $\mathbb{P}^2$ and $f_{p,l} : X_{p,l} \dashrightarrow Z_{p.l}$ is the map that "takes pictures":

$$\begin{aligned}\Big(X_1, \ldots, X_p, \overline{L_1\, L_1'}, \ldots, \overline{L_l\, L_l'}, [\mathbf{R} \mid \mathbf{t}]\Big) \mapsto \\ \Big(X_1, \ldots, X_p, \overline{L_1\, L_1'}, \ldots, \overline{L_l\, L_l'}, [\mathbf{R} \mid \mathbf{t}]\, X_1, \ldots, [\mathbf{R} \mid \mathbf{t}]\, X_p, \\ \overline{[\mathbf{R} \mid \mathbf{t}]\, L_1\ [\mathbf{R} \mid \mathbf{t}]\, L_1'}, \ldots, \overline{[\mathbf{R} \mid \mathbf{t}]\, L_l\ [\mathbf{R} \mid \mathbf{t}]\, L_l'}\Big)\end{aligned}$$

(here $\overline{L\, L'}$ is the line spanned by L and L'.) Counting dimensions gives $\dim X_{p,l} = 3(p + l) + 6$ and $\dim Z_{p,l} = 5(p + l)$. Equating the two, we see that the only possibilities are $(p, l) = (3, 0), (2, 1), (1, 2), (0, 3)$. The first case corresponds to the P3P problem.

Result 1. The full list of Galois/monodromy groups of branched covers $f_{p,l}$ is as follows:

$$\begin{aligned}\mathrm{Mon}(X_{3,0}/Z_{3,0}) &\cong S_2 \wr S_4 \cap A_8 \hookrightarrow S_8 \\ \mathrm{Mon}(X_{2,1}/Z_{2,1}) &\cong S_2 \wr S_2 \cap A_4 \cong C_2 \times C_2 \hookrightarrow S_4 \\ \mathrm{Mon}(X_{1,2}/Z_{1,2}) &\cong S_2 \wr S_4 \cap A_8 \hookrightarrow S_8 \\ \mathrm{Mon}(X_{0,3}/Z_{0,3}) &\cong S_8.\end{aligned}$$

We note the respective degrees $8,\ 4,\ 8,\ 8$ agree with those reported in [93], in which these problems were formulated using different systems of equations. The systems of equations defining the parameter homotopies used for Result 1 were constructed as follows:

- Points in the world are represented by 4×1 matrices $X_1, \ldots, X_p$.
- Points in the image are represented by 3×1 matrices $x_1, \ldots, x_p$.
- World lines are kernels of 2×4 matrices $[\mathbf{N}_1 \mid \mathbf{N}_1']^\top, \ldots, [\mathbf{N}_l \mid \mathbf{N}_l']^\top$.
- Image lines are kernels of 1×3 matrices $\mathbf{n}_1^\top, \ldots \mathbf{n}_l^\top$.

- We enforce rank constraints by the vanishing of maximal minors of certain matrices:
  - point-to-point: $\operatorname{rk}\left([\mathbf{R} \mid \mathbf{t}]\, X_i \mid x_i\right) \leqslant 1$ for $i = 1, \ldots, p$
  - line-to-line: $\operatorname{rk}\left(\mathbf{N}_i \mid \mathbf{N}'_i \mid [\mathbf{R} \mid \mathbf{t}]^\top \mathbf{n}_i\right) \leqslant 2$ for $i = 1, \ldots, l$
- A well-constrained system is extracted with Algorithm 3.

The case $(p, l) = (3, 0)$ reduces to solving the P3P problem as formulated in Equation 4.10. The literature on this problem is vast, and the earliest work [49] pre-dates the field of computer vision by more than a century. The degree of this problem is 8 and the Galois/monodromy group is a subgroup of $S_2 \wr S_4$ due to the sign symmetry. In the terminology of Brysiewicz et al. [24], Equation 4.10 defines a *lacunary* polynomial system whose monomial supports span a proper sublattice of $\mathbb{Z}^3$ with finite index. In the setting of that paper, we would consider the family of all systems with the same monomial supports as in Equation 4.10

$$\begin{aligned} h_{1,2} &= Ax_1^2 + Bx_2^2 + Cx_1x_2 + D \\ h_{1,3} &= Ex_1^2 + Fx_3^2 + Gx_1x_3 + H \\ h_{2,3} &= Ix_2^2 + Jx_3^2 + Kx_2x_3 + L \end{aligned} \tag{4.11}$$

This gives a branched cover $X_h \to \mathbb{C}^{12}$ where $X_h = V(h_{1,2}, h_{1,3}, h_{2,3}) \subset \mathbb{C}^3 \times \mathbb{C}^{12}$. On the other hand, for P3P the natural branched cover is $X_f \to \mathbb{C}^6$, where $X_f \subset \mathbb{C}^3 \times \mathbb{C}^6$. We find numerically that $\operatorname{Mon}(X_h/\mathbb{C}^{12})$ is the full wreath product $S_2 \wr S_4$, whereas our numerical experiments suggest that the Galois/monodromy group for P3P is the *proper subgroup* $S_2 \wr S_4 \cap A_8$.

To certify the result of our numerical monodromy computation, we can compute the Galois group for P3P using symbolic computation. Consider $I = \langle f_{1,2}, f_{1,3}, f_{2,3} \rangle$ as an ideal in a polynomial ring $\mathbb{F}[x_1, x_2, x_3]$ whose coefficient field is $\mathbb{F} = \mathbb{C}(Z) = \mathbb{C}(\vec{c}, \vec{d})$. The dimension and degree of I are 0 and 8. We can compute a lexicographic Gröbner basis for I with $x_1 > x_2 > x_3$ in a matter of seconds using the FGLM algorithm [41], implemented for Macaulay2 [46] in the package `FGLM` [91]. The Gröbner basis $G = \{g_1, g_2, g_3\}$ has the

form predicted by the Shape lemma:

$$\begin{aligned} g_1(x_1, x_2, x_3) &= x_1 + r_1(\vec{c}, \vec{d})\, x_3 \\ g_2(x_2, x_3) &= x_2 + r_2(\vec{c}, \vec{d})\, x_3 \\ g_3(x_3) &= x_3^8 + A(\vec{c}, \vec{d})\, x_3^6 + B(\vec{c}, \vec{d})\, x_3^4 + C(\vec{c}, \vec{d})\, x_3^2 + D(\vec{c}, \vec{d}), \end{aligned}$$

for particular rational functions $r_1, r_2, A, B, C, D \in \mathbb{F}$. We see that x_3 is a primitive element for the extension $\mathbb{C}(X)/\mathbb{C}(Z)$. To verify that $\mathrm{Mon}(X/Z) \cong \mathrm{Gal}(X/Z)$ is contained in A_8, it suffices to show that the discriminant of g_3 is square. For arbitrary coefficients (A, B, C, D), the discriminant of $x_3^8 + A\, x_3^6 + B\, x_3^4 + C\, x_3^2 + D$ is the product of D and a square. For P3P, $D(\vec{c}, \vec{d})$ is also a square.

Factorizations of P3P are also classical: from Equation 4.10, we have

$$\begin{aligned} y_1(1 + y_2^2 - c_{1,2}y_2) - d_{1,2}^2 &= 0 \\ y_1(1 + y_3^2 - c_{1,3}y_3) - d_{1,3}^2 &= 0 \\ y_1(y_2^2 + y_3^2 - c_{2,3}y_2y_3) - d_{2,3}^2 &= 0 \end{aligned} \tag{4.12}$$

where $y_1 = x_1^2$, $y_2 = x_2/x_1$, $y_3 = x_3/x_1$ are separating invariants [67] for the action of the deck transformation group. We see that even when $X \to Z$ is regular in the definition of a factorization as in Equation 4.4, the maps $X \dashrightarrow Y$ and $Y \dashrightarrow Z$ need not be.

In contrast to the case of 3 points, the branched cover $X_{0,3} \dashrightarrow Z_{0,3}$ corresponding to the case of 3 lines is indecomposable, since its Galois/monodromy group S_8 acts primitively on the set of 8 solutions. Thus, any algebraic algorithm for solving this problem must be capable of computing the roots of a polynomial of degree 8 or higher.

We now consider the more interesting "mixed cases" $(p, l) \in \{(2, 1), (1, 2)\}$. Proposition 4.1.4 shows that each of the mixed cases has a nontrivial deck transformation group: we have that $\mathrm{Aut}(X_{2,1}/Z_{2,1}) \cong C_2 \times C_2$ and $\mathrm{Aut}(X_{1,2}/Z_{1,2}) \cong C_2$. Using the rank constraints described above, we were able to observe numerically that solutions in the same block for both of these mixed cases differed by a reflection. These deck transformations take on a particularly simple form after changing coordinates as in [93].

For the case $(p, l) = (2, 1)$, the formulation [93, Equations 4,5] makes use of a clever choice of reference frames to get equations

$$\begin{aligned} A\,\boldsymbol{X} - b &= 0 \\ R_{1,1}^2 + R_{2,1}^2 + R_{3,1}^2 - 1 &= 0 \\ R_{2,1}^2 + R_{2,2}^2 + R_{2,3}^2 - 1 &= 0 \end{aligned} \tag{4.13}$$

where A and b are 6×8 and 8×1 matrices depending on the given data, and

$$\boldsymbol{X} = [R_{1,1}, R_{2,1}, R_{3,1}, R_{2,2}, R_{2,3}, t_1, t_2, t_3]^\top$$

is a vector of indeterminates. Using `FGLM` as in the previous section, we discover new constraints

$$\begin{aligned} R_{3,1}^2 + \psi_1(A, b) &= 0 \\ t_3^2 + 2t_3 + \psi_2(A, b) &= 0, \end{aligned}$$

for particular rational functions ψ_1, ψ_2 in the data, which did not appear in Equation 4.13 originally. Formulas for the deck transformations of this Galois cover follow by way of the basic Example 9. The remaining constraints output by `FGLM` are, as expected, of the form

$$\begin{aligned} R_{i,j} + \ell_{i,j}(R_{3,1}) &= 0 \\ t_j + \ell_j(t_3) &= 0 \end{aligned}$$

for linear forms $\ell_j, \ell_{i,j}$ over the coefficient field $\mathbb{Q}(A, b)$. For this very special problem, the Gröbner basis elements are surprisingly compact. This suggests, as an alternative to the solution proposed in [93], that we may solve for the rotation and translation independently.

Likewise, for the $(p, l) = (1, 2)$ case, using the similar formulation of [93, Equations 7,8], we discover the following symmetry in the solutions ($\mathbf{e}_3 \in \mathbb{R}^3$ is the third standard basis vector):

$$(\vec{R},\, \mathbf{t}) \mapsto (-\vec{R}, -\mathbf{t} - 2\mathbf{e}_3).$$

We note that in this formulation, $\vec{R}$ contains only the first two rows of the unknown rotation matrix. In hindsight, this symmetry is quite easy to verify. However, we stress that computing the Galois/monodromy group was what led us to discover it.

4.2.2 Relative pose of points and lines

We now consider Galois/monodromy groups of problems discussed in Chapter 3. Here, the problems have much larger degree, making them well-suited for numerical methods.

Result 2. Among all minimal problems of degree $< 1,000$ appearing in Table 3.1, all have either an imprimitive or full symmetric Galois/monodromy group. The imprimitive cases are:

$$\begin{aligned}
\mathrm{Mon}\left(\boxed{\cdot\cdot}\right) &\cong (C_2)^2 \rtimes (S_2 \wr S_3 \cap A_6) \hookrightarrow S_{12} \\
\mathrm{Mon}\left(\boxed{\cdot\cdot}\right) &\cong S_2 \wr S_8 \cap A_{16} \hookrightarrow S_{16} \\
\mathrm{Mon}\left(\boxed{\cdot\cdot}\right) &\cong S_2 \wr S_{10} \cap A_{20} \hookrightarrow S_{20} \\
\mathrm{Mon}\left(\boxed{\cdot\cdot}\right) &\cong S_2 \wr (S_2 \wr S_{16} \cap A_{32}) \cap A_{64} \hookrightarrow S_{64} \\
\mathrm{Mon}\left(\boxed{\cdot\cdot}\right) &\cong (C_2)^4 \rtimes \left((C_2)^4 \rtimes (S_2 \wr (S_2 \wr S_4))\right) \hookrightarrow S_{64} \\
\mathrm{Mon}\left(\boxed{\cdot\cdot}\right) &\cong (C_2)^2 \rtimes \left(C_2^2 \rtimes (S_2 \wr (S_2 \wr S_2 \cap A_4) \cap A_8)\right) \hookrightarrow S_{32}.
\end{aligned}$$

For the sake of uniformity, we have used the semidirect product $\rtimes$ to indicate subgroups of an appropriate wreath product. Thus, for instance, for $\mathrm{Mon}\left(\boxed{\cdot\cdot}\right)$, the outermost $(C_2)^2$ should be regarded as a subgroup of $(S_2)^{16}$, and the innermost as a subgroup of $(S_2)^8$. Much to our surprise, the group $\mathrm{Mon}\left(\boxed{\cdot\cdot}\right)$ turns out to be solvable. The fact that $\mathrm{Mon}\left(\boxed{\cdot\cdot}\right) \cong S_{312}$ shows that the homotopy solver developed in [40] is *optimal* in the sense of tracking the fewest paths possible. The problem appearing in [35] can be thought of as P3P fibered over the five-point problem. Unlike the majority of problems studied here, this composite minimal problem has an intermediate field $\mathbb{C}(Z) \subsetneq \mathbb{C}(Y) \subsetneq \mathbb{C}(X)$ which is not the fixed field of some subgroup of $\mathrm{Aut}(X/Z)$. This can be seen by comparing the lattice of block systems with the subgroup lattice of $\mathrm{Aut}(X/Z)$.

In the remainder of this section, we describe explicit decompositions of two problems

in Result 2—the problem ⚄, which is a degenerate case of the five-point problem, and the problem ⚃, which is a degenerate case of the notorious "four points in three views" problem for calibrated cameras.

For ⚃, we use similar notation as in the five-point problem, with constraints

$$\begin{aligned} \mathbf{R}^\top \mathbf{R} = \mathbf{I}, \quad \det \mathbf{R} = 1, \\ \beta_i \mathbf{y}_i = \mathbf{R}\alpha_i \mathbf{x}_i + \mathbf{t}, \quad \alpha_i, \beta_i \neq 0, \quad i = 1, \ldots, 4, \\ \det\left(\begin{bmatrix} \alpha_1\mathbf{x}_1 & \alpha_2\mathbf{x}_2 & \alpha_3\mathbf{x}_3 & \alpha_4\mathbf{x}_4 \\ 1 & 1 & 1 & 1 \end{bmatrix}\right) = 0. \end{aligned} \tag{4.14}$$

Our branched cover has a base space

$$Z = \left(\mathbb{C}^2 \times \{1\}\right)^4 \times \left(\mathbb{C}^2 \times \{1\}\right)^4$$

and its total space X is given by

$$X \subseteq \mathrm{SO}_{\mathbb{C}}(3) \times \mathbb{P}^{10}_{\mathbb{C}} \times$$

such that Equation 4.14 holds for all $(\mathbf{R}, (\mathbf{t}, \alpha_1, \ldots, \alpha_4, \beta_1, \ldots, \beta_4), z) \in X$. Projection of X onto Z defines a branched cover of degree 12. This branched cover is birationally equivalent to the joint camera map of the problem ⚃ from [34], since the fifth point on both lines in each image is generically determined from the other four points. Result 2 tells us the Galois/monodromy group is $C_2 \times C_2 \rtimes (S_2 \wr S_3 \cap A_6)$. The GAP command `MinimalGeneratingSet` shows that this group is minimally generated by two permutations: in cycle notation,

$$\mathrm{Mon}(X/Z) \cong \Big\langle (1\ \ 2)(3\ \ 4)(5\ \ 12\ \ 8\ \ 9)(6\ \ 11\ \ 7\ \ 10),\ (1\ \ 11\ \ 5)(2\ \ 10\ \ 8)(3\ \ 9\ \ 7)(4\ \ 12\ \ 6) \Big\rangle. \tag{4.15}$$

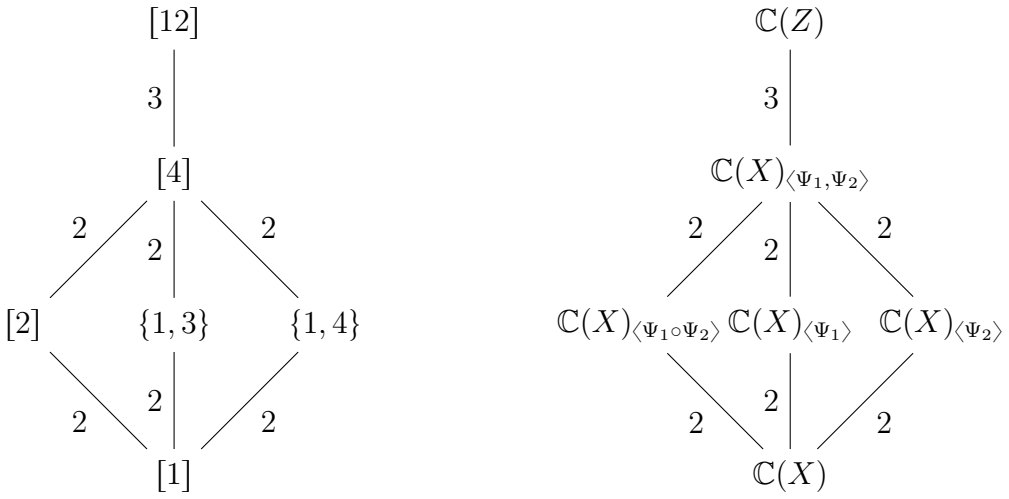


Figure 4.4: Correspondence between block systems (left) and intermediate fields (right) for the calibrated homography problem. The notation K_H means the intermediate field of an extension K/F fixed elementwise by a subgroup $H \leqslant \mathrm{Aut}(K/F)$.

The lattice of block systems is depicted on the left in Figure Figure 4.4. The vertex labels correspond to stabilizer subgroups of $\mathrm{Mon}(X/Z)$, and the edges are labeled by the degrees of maps appearing in some decomposition of the form in Equation 4.9. To the right is the inverted lattice of intermediate fields. Like the majority of examples in this paper, $\mathbb{C}(X)/\mathbb{C}(Z)$ is not a Galois extension.

Before we determine a decomposition, we first describe the group of deck transformations. The centralizer in S_{12} is

$$\Big\langle (1\ \ 3)(2\ \ 4)(5\ \ 7)(6\ \ 8)(9\ \ 11)(10\ \ 12),\ (1\ \ 4)(2\ \ 3)(5\ \ 8)(6\ \ 7)(9\ \ 12)(10\ \ 11) \Big\rangle \cong C_2 \times C_2.$$

The deck transformation corresponding to the first generator is the twisted pair map Ψ_1, defined just as in Equation 4.5. The second is a reflection-rotation symmetry Ψ_2 depicted in Figure 4.5. To get a formula for Ψ_2, it is convenient to work with the equation of the unknown plane:

$$\langle \mathbf{n}, \boldsymbol{X} \rangle = d. \tag{4.16}$$

Note that $\mathbf{n}$ and d depend rationally on the data. The formula for Ψ_2 is given by

$$\begin{aligned}
\Psi_2(\mathbf{R}) &= \mathbf{R}\left(2\frac{\mathbf{n}\mathbf{n}^\top}{\mathbf{n}^\top\mathbf{n}} - \mathbf{I}\right) \\
\Psi_2(\mathbf{t}) &= -\mathbf{t} - \frac{2d}{\mathbf{n}^\top\mathbf{n}}\mathbf{R}\mathbf{n} \\
\Psi_2(\alpha_i) &= \alpha_i \\
\Psi_2(\beta_i) &= -\beta_i \\
\Psi_2(\mathbf{x}_1, \ldots, \mathbf{x}_4, \mathbf{y}_1, \ldots, \mathbf{y}_4) &= (\mathbf{x}_1, \ldots, \mathbf{x}_4, \mathbf{y}_1, \ldots, \mathbf{y}_4).
\end{aligned} \tag{4.17}$$

To better understand the effect of Ψ_2 on $\mathbf{t}$, let $\boldsymbol{X}$ be any point on the scene plane and calculate

$$-\mathbf{t} - \frac{2d}{\mathbf{n}^\top\mathbf{n}}\mathbf{R}\,\mathbf{n} = -\mathbf{t} - \mathbf{R}\boldsymbol{X} - \mathbf{R}\left(2\frac{\mathbf{n}\mathbf{n}^\top}{\mathbf{n}^\top\mathbf{n}} - \mathbf{I}\right)\boldsymbol{X} \tag{4.18}$$

$$= -\mathbf{R}\left(2\frac{\mathbf{n}\mathbf{n}^\top}{\mathbf{n}^\top\mathbf{n}} - \mathbf{I}\right)\left(\left(\mathbf{I} - 2\frac{\mathbf{n}\mathbf{n}^\top}{\mathbf{n}^\top\mathbf{n}}\right)(-\mathbf{R}^\top\mathbf{t} - \boldsymbol{X}) + \boldsymbol{X}\right) \tag{4.19}$$

$$= -\Psi_2(\mathbf{R})\left(\left(\mathbf{I} - 2\frac{\mathbf{n}\mathbf{n}^\top}{\mathbf{n}^\top\mathbf{n}}\right)(-\mathbf{R}^\top\mathbf{t} - \boldsymbol{X}) + \boldsymbol{X}\right) \tag{4.20}$$

This may be understood as follows: we take $-\mathbf{R}^\top\mathbf{t}$, which is the center of the second camera expressed in the frame of the first camera (cf. Eq. Equation 4.14), then reflect this vector through the plane and transform it back to the vector representing the center of the first camera expressed in the frame of the reflected second camera (by multiplying by $-\Psi_2(\mathbf{R})$).

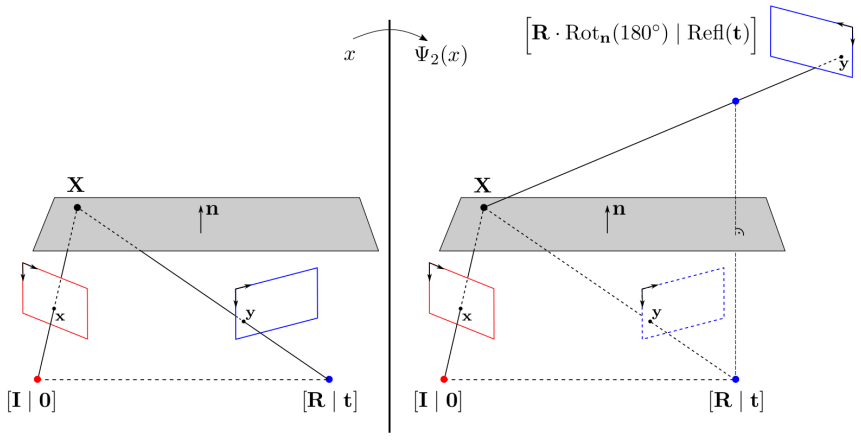


Figure 4.5: Reflection-rotation symmetry for Equation 4.14

Proposition 4.2.1. Ψ_1 and Ψ_2 generate the deck transformation group for the planar calibrated homography problem $X \to Z$ defined by Equation 4.14. The corresponding permutations which centralize $\mathrm{Mon}(X/Z)$ are as follows:

$$\begin{aligned}
\Psi_1 \circ \Psi_2 &\leftrightarrow (1\ 2)(3\ 4)(5\ 6)(7\ 8)(9\ 10)(11\ 12) \\
\Psi_1 &\leftrightarrow (1\ 3)(2\ 4)(5\ 7)(6\ 8)(9\ 11)(10\ 12) \\
\Psi_2 &\leftrightarrow (1\ 4)(2\ 3)(5\ 8)(6\ 7)(9\ 12)(10\ 11)
\end{aligned}$$

These correspond to the maximal chains in the lattice of block systems (see Figure 4.4).

Proof. We verify that Ψ_1 and Ψ_2 really are deck transformations. The rest is elementary or follows from Proposition 4.1.4. Appealing to well-known properties of the twisted pair Ψ_1, it suffices for us to check that planarity of the scene is preserved:

$$\det\left(\begin{bmatrix} \Psi_1(\alpha_1)\mathbf{x}_1 & \Psi_1(\alpha_2)\mathbf{x}_2 & \Psi_1(\alpha_3)\mathbf{x}_3 & \Psi_1(\alpha_4)\mathbf{x}_4 \\ 1 & 1 & 1 & 1 \end{bmatrix}\right) = 0.$$

Letting $\mathbf{m} = \frac{2}{\mathbf{t}^\top\mathbf{t}}\mathbf{R}^\top\mathbf{t}$, we may compute this determinant as follows:

$$\left(\prod_{i=1}^{4}(1+\mathbf{m}^\top\alpha_i\mathbf{x}_i)\right)^{-1}\det\left(\begin{bmatrix} \alpha_1\mathbf{x}_1 & \alpha_2\mathbf{x}_2 & \alpha_3\mathbf{x}_3 & \alpha_4\mathbf{x}_4 \\ 1+\mathbf{m}^\top\alpha_1\mathbf{x}_1 & 1+\mathbf{m}^\top\alpha_2\mathbf{x}_2 & 1+\mathbf{m}^\top\alpha_3\mathbf{x}_3 & 1+\mathbf{m}^\top\alpha_4\mathbf{x}_4 \end{bmatrix}\right) =$$
$$\left(\prod_{i=1}^{4}(1+\mathbf{m}^\top\alpha_i\mathbf{x}_i)\right)^{-1}\det\left(\begin{bmatrix} \alpha_1\mathbf{x}_1 & \alpha_2\mathbf{x}_2 & \alpha_3\mathbf{x}_3 & \alpha_4\mathbf{x}_4 \\ 1 & 1 & 1 & 1 \end{bmatrix}\right) = 0.$$

For Ψ_2, it is clear that planarity of the scene is preserved and that $\Psi_2(\mathbf{R}) \in \mathrm{SO}_{\mathbb{C}}(3)$. If we substitute $\Psi_2(x)$ into the point correspondence constraint in Equation 4.14 and take $\boldsymbol{X} = \alpha_i\,\mathbf{x}_i$ in Equation 4.18, then

$$-\beta_i\mathbf{y}_i = \mathbf{R}\left(2\frac{\mathbf{n}\mathbf{n}^\top}{\mathbf{n}^\top\mathbf{n}} - \mathtt{I}\right)\alpha_i\mathbf{x}_i + \left(-\mathbf{t} - \mathbf{R}\alpha_i\mathbf{x}_i - \mathbf{R}\left(2\frac{\mathbf{n}\mathbf{n}^\top}{\mathbf{n}^\top\mathbf{n}} - \mathtt{I}\right)\alpha_i\mathbf{x}_i\right) = -\mathbf{R}\alpha_i\mathbf{x}_i - \mathbf{t}.$$

We conclude that Equation 4.14 is invariant up to sign under application of Ψ_2. □

Finally, we describe a decomposition of $X \to Z$

$$X \dashrightarrow Y_1 \dashrightarrow Y_2 \to Z, \tag{4.21}$$

corresponding to the left-most chain in Figure 4.4. This decomposition makes use of the *calibrated homography matrix* associated to $(\mathbf{R}, \mathbf{t})$ and the scene plane:

$$\mathbf{H} = \mathbf{R} + \frac{1}{d}\mathbf{t}\mathbf{n}^\top. \tag{4.22}$$

Up to scale, any 3×3 matrix has the form Equation 4.22. On the other hand, any real calibrated homography matrix has an eigenvalue equal to 1 (see eg. [83, Lemma 5.18]), and thus lies on an irreducible hypersurface of degree 6:

$$\mathcal{H}_1 = \{\mathbf{H} \in \mathbb{C}^{3\times 3} \mid \det(\mathbf{H}^\top\mathbf{H} - \mathbf{I}) = 0\}.$$

In our decomposition, we may take

$$Y_1 = \{(\mathbf{H}, ((\mathbf{x}_1, \ldots, \mathbf{x}_4), (\mathbf{y}_1, \ldots, \mathbf{y}_4))) \in \mathcal{H}_1 \times \left(\mathbb{P}^2\right)^4 \times \left(\mathbb{P}^2\right)^4 \mid \mathbf{x}_i \sim \mathbf{H}\mathbf{y}_i,\ i = 1, \ldots 4\}.$$

Here we use the standard notation $\sim$ indicate that two vectors are equal up to scale. We note that each of these correspondence constraints is equivalent to the vanishing of three homogeneous, non-independent linear equations

$$\left[\mathbf{x}_i\right]_\times \mathbf{H}\mathbf{y}_i = 0.$$

A short calculation reveals that $x \in X_z$ and $\Psi_1 \circ \Psi_2(x) \in X_z$ map to the same point in Y_1. We also note that Y_1 is irreducible, since its Zariski-open in the graph of $\mathcal{H}_1 \times \left(\mathbb{P}^2\right)^4 \dashrightarrow \left(\mathbb{P}^2\right)^4$.

The projection $Y_1 \to Z$ has a deck transformation given by the sign-symmetry $\mathbf{H} \mapsto -\mathbf{H}$. To remove this last symmetry, we define

$$\begin{aligned} s &= \frac{1}{\mathbf{H}_{1,1}^2} \\ \mathbf{S} &= \frac{1}{\mathbf{H}_{1,1}}\mathbf{H} \end{aligned} \tag{4.23}$$

and take $Y_1 \dashrightarrow Y_2 \subset \mathbb{C}^9 \times \left(\mathbb{P}^2\right)^4 \times \left(\mathbb{P}^2\right)^4$ by mapping $\mathbf{H}$ to s and the 8 non-constant entries

of $\mathbf{S}$. Algebraically, the ideal

$$\langle \det(\mathbf{S}^\top \mathbf{S} - s\mathbf{I}),\ \left[\mathbf{x}_1\right]_\times \mathbf{S}\mathbf{y}_1,\ \left[\mathbf{x}_2\right]_\times \mathbf{S}\mathbf{y}_2,\ \left[\mathbf{x}_3\right]_\times \mathbf{S}\mathbf{y}_3,\ \left[\mathbf{x}_4\right]_\times \mathbf{S}\mathbf{y}_4 \rangle \tag{4.24}$$

has dimension 0 and degree $3 = \deg(Y_2/Z)$ for generic data $(\mathbf{x}_1, \ldots, \mathbf{y}_4) \in Z$. The algebraic complexity as captured by the Galois group matches that of a well-known algorithm for computing $\mathbf{H}$, in which one must compute the singular values of a 3×3 matrix $\lambda\mathbf{H}$ recovered up to scale from the four point correspondences (see eg. [53, Algorithm 4.1] or [83, Algorithm 5.2].)

Finally, we consider the minimal problem  , where the task is to recover the relative orientation of three cameras from the input data of four point correspondences which lie on the incidence variety

$$Z = \{(\mathbf{x}_1, \ldots, \mathbf{x}_4, \mathbf{y}_1, \ldots, \mathbf{y}_4, \mathbf{z}_1, \ldots, \mathbf{z}_4) \in \left(\mathbb{P}^2\right)^{12} \mid \mathbf{x}_3 \in \overline{\mathbf{x}_1\,\mathbf{x}_2},\ \mathbf{y}_3 \in \overline{\mathbf{y}_1\,\mathbf{y}_2},\ \mathbf{z}_3 \in \overline{\mathbf{z}_1\,\mathbf{z}_2}\}.$$

Unlike the two-view problem  , the three-view problem no longer has a twisted pair symmetry. However, the deck transformation group for is $C_2 \times C_2$, generated by two deck transformations analagous to $\Psi_1 \circ \Psi_2$ in the two-view case. For , the joint camera map defined in section 3.2 is birationally equivalent to a branched cover whose fibers are pairs of homography matrices which are *compatible* in the sense that they share the same normal vector. Thus, the solutions of interest lie on the subvariety $\mathcal{H}_2 \subset \left(\mathbb{C}^{3\times 3}\right)^2$ defined to be the closed image of the map

$$\begin{aligned} \left(\mathrm{SO}_{\mathbb{C}}(3)\right)^2 \times \left(\mathbb{C}^3\right)^3 &\to \left(\mathbb{C}^{3\times 3}\right)^2 \\ (\mathbf{R}_1, \mathbf{R}_2, \mathbf{t}_1, \mathbf{t}_2, \mathbf{n}) &\mapsto (R_1 + \mathbf{t}_1\mathbf{n}^\top, R_2 + \mathbf{t}_2\mathbf{n}^\top). \end{aligned}$$

Notice that, unlike in Equation 4.22, we have absorbed the constant $\frac{1}{d}$ into $\mathbf{t}$ for each homography matrix. We wish to compute the fibers of the branched cover $X \to Z$, where

$$X = \{((\mathbf{H}_1, \mathbf{H}_2), (\mathbf{x}_1, \ldots, \mathbf{z}_4)) \in \mathcal{H}_2 \times Z \mid \mathbf{H}_1\,\mathbf{x}_i \sim \mathbf{y}_i,\ \mathbf{H}_2\,\mathbf{x}_i \sim \mathbf{z}_i,\ i = 1, \ldots 4\}. \tag{4.25}$$

For this problem, we have $\deg(X/Z) = 64$, and Result 2 tells us $\mathrm{Mon}(X/Z) \cong S_2 \wr (S_2 \wr S_{16} \cap A_{32}) \cap A_{64}$. It follows that there exists a decomposition

$$X \dashrightarrow Y_1 \dashrightarrow Y_2 \dashrightarrow Z$$

with $\deg(X/Y_1) = \deg(Y_1/Y_2) = 2$ and $\deg(Y_2/Z) = 16$. The deck transformations of $X \to Z$ are easily seen to be $(\mathbf{H}_1, \mathbf{H}_2) \mapsto (\pm\mathbf{H}_1, \pm\mathbf{H}_2)$. Thus, we may use separating invariants $s_1, \mathbf{S}_1, s_2, \mathbf{S}_2$ as in the two view case to write down the maps $X \dashrightarrow Y_1$ and $Y_1 \dashrightarrow Y_2$.

However, our description of X is unsatisfying from the point of view of constructing a polynomial solvers, since we have only described $\mathcal{H}_2$ parametrically. We leave determining the ideal $\mathcal{I}_{\mathcal{H}_2}$ as a challenging open problem in algebraic vision, analogous to previous works [3, 2]. Our final Result 3 is a partial solution to this implicitization problem, which describes an ideal contained in $\mathcal{I}_{\mathcal{H}_2}$. The generators of this ideal and the linear correspondence constraints in Equation 4.25 generate a 0-dimensional ideal of degree 64 for generic data $z = (\mathbf{x}_i, \mathbf{y}_i, \mathbf{z}_i)$.

Drawing on the description of $\mathcal{H}_1$ from the previous section, consider the map

$$\begin{aligned} \mathcal{H}_2 &\to \left(\mathbb{C}^{3\times 3}\right)^2 \\ (\mathbf{H}_1, \mathbf{H}_2) &\mapsto (\mathbf{H}_1^\top \mathbf{H}_1 - \mathbf{I}, \mathbf{H}_2^\top \mathbf{H}_2 - \mathbf{I}). \end{aligned}$$

The image of this map has the alternate parametrization

$$\begin{aligned} \left(\mathbb{C}^{3\times 1}\right)^3 &\to \left(\mathbb{C}^{3\times 3}\right)^2 \\ (\mathbf{d}_1, \mathbf{d}_2, \mathbf{n}) &\mapsto (\mathbf{n}\mathbf{d}_1^\top + \mathbf{d}_1\mathbf{n}^\top, \mathbf{n}\mathbf{d}_2^\top + \mathbf{d}_2\mathbf{n}^\top). \end{aligned}$$

Using Macaulay2, we compute implicit equations in new matrix variables $\mathbf{W}_i = \mathbf{n}\mathbf{d}_i^\top + \mathbf{d}_i\mathbf{n}^\top$, $i = 1, 2$. The resulting elimination ideal in $\mathbb{C}[\mathbf{W}_1, \mathbf{W}_2]$ is generated by four cubics and 15 quartics. The cubic constraints obtained are

$$\det(\mathbf{W}_1) = \det(\mathbf{W}_2) = \det(\mathbf{W}_1 + \mathbf{W}_2) = \det(\mathbf{W}_1 - \mathbf{W}_2) = 0. \tag{4.26}$$

These cubics can be understood in terms of the alternate parametrization, which shows that generic $(\mathbf{W}_1, \mathbf{W}_2)$ in the image will span a pencil of rank-2 symmetric matrices. In what in follows, it is enough for us to consider two of the 15 quartics, which have alternate expressions in terms of resultants:

$$\begin{aligned} \mathrm{Res}_{n_1}\left(\mathbf{W}_1^{3,3} n_1^2 - 2\mathbf{W}_1^{1,3} n_1 + \mathbf{W}_1^{1,1}, \mathbf{W}_2^{3,3} n_1^2 - 2\mathbf{W}_2^{1,3} n_1 + \mathbf{W}_2^{1,1}\right) = 0 \\ \mathrm{Res}_{n_2}\left(\mathbf{W}_1^{3,3} n_2^2 - 2\mathbf{W}_1^{2,3} n_2 + \mathbf{W}_1^{2,2}, \mathbf{W}_2^{3,3} n_2^2 - 2\mathbf{W}_2^{2,3} n_2 + \mathbf{W}_2^{2,2}\right) = 0 \end{aligned} \tag{4.27}$$

where $\mathbf{n} = (n_1, n_2, n_3)$.

Substituting $\mathbf{W}_i = \mathbf{H}_i^\top \mathbf{H}_i - \mathbf{I}$ into Equation 4.26 yields four polynomials of degree 6 vanishing on $\mathcal{H}_2$. Using Bertini [15], we computed points where these equations vanish by tracking $6^4 = 1296$ homotopy paths. Out of these points, 336 lie on $\mathcal{H}_2$. The remaining 960 paths resulted in 224 points not on $\mathcal{H}_2$, each occurring with multiplicity 4. We confirmed that the degree of the variety $\mathcal{H}_2$ is indeed 336 using monodromy.

Unlike in the five-point problem, the linear equations implied by $\mathbf{H}_i \mathbf{x}_i \sim \mathbf{y}_i$ and $\mathbf{H}_2 \mathbf{x}_i \sim \mathbf{z}_i$ are non-generic. The number of solutions to these linear equations and the four degree-6 equations obtained from Equation 4.26 is 320. Out of these solutions, only 84 satisfy the degree-8 equations obtained from Equation 4.27. To obtain the degree 64 reported in [34], it is sufficient to impose the additional constraint $\det \mathbf{H}_1 \neq 0$. In summary, we have the following result.

Result 3. For each $z = (\mathbf{x}_1, \ldots, \mathbf{z}_4) \in Z$, let $I_z \subset \mathbb{C}[\mathbf{H}_1, \mathbf{H}_2, D]$ be the ideal in 19 variables generated by the linear relations $\mathbf{H}_1 \mathbf{x}_i \sim \mathbf{y}_i$, $\mathbf{H}_2 \mathbf{x}_i \sim \mathbf{z}_i$ for $i = 1, \ldots, 4$, the saturation constraint $D \det \mathbf{H}_1 - 1 = 0$, and six equations obtained by setting $(\mathbf{W}_1, \mathbf{W}_2) = (\mathbf{H}_1^\top \mathbf{H}_1 - \mathbf{I}, \mathbf{H}_2^\top \mathbf{H}_2 - \mathbf{I})$ in Equation 4.26 and Equation 4.27. For generic $z \in Z$, we have $\deg(I_z) = 64$. Moreover, after substituting $s_1, s_2, \mathbf{S}_1, \mathbf{S}_2$ as in Equation 4.23, Equation 4.24, we obtain an ideal of degree 16 for generic data.

BIBLIOGRAPHY

[1] Sameer Agarwal, Yasutaka Furukawa, Noah Snavely, Ian Simon, Brian Curless, Steven M Seitz, and Richard Szeliski. Building Rome in a day. *Communications of the ACM*, 54(10):105–112, 2011.

[2] Sameer Agarwal, Andrew Pryhuber, and Rekha R. Thomas. Ideals of the multiview variety. *IEEE Trans. Pattern Anal. Mach. Intell.*, 43(4):1279–1292, 2021.

[3] Chris Aholt and Luke Oeding. The ideal of the trifocal variety. *Math. Comp.*, 83(289):2553–2574, 2014.

[4] Cenek Albl, Zuzana Kukelova, and Tomas Pajdla. R6P-rolling shutter absolute camera pose. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2292–2300, 2015.

[5] Eugene L. Allgower and Kurt Georg. *Introduction to numerical continuation methods*, volume 45 of *Classics in Applied Mathematics*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2003. Reprint of the 1990 edition [Springer-Verlag, Berlin; MR1059455 (92a:65165)].

[6] E. Arbarello, M. Cornalba, P. A. Griffiths, and J. Harris. *Geometry of algebraic curves. Vol. I*, volume 267 of *Grundlehren der Mathematischen Wissenschaften [Fundamental Principles of Mathematical Sciences]*. Springer-Verlag, New York, 1985.

[7] Emil Artin. *Galois theory*. Dover Publications, Inc., Mineola, NY, second edition, 1998. Edited and with a supplemental chapter by Arthur N. Milgram.

[8] Erik Ask, Yubin Kuang, and Kalle Åström. Exploiting p-fold symmetries for faster polynomial equation solving. In *Proceedings of the 21st International Conference on Pattern Recognition, ICPR 2012, Tsukuba, Japan, November 11-15, 2012*, pages 3232–3235. IEEE Computer Society, 2012.

[9] Chad Awtrey, Nakhila Mistry, and Nicole Soltz. Centralizers of transitive permutation groups and applications to Galois theory. *Missouri J. Math. Sci.*, 27(1):16–32, 2015.

[10] László Babai. The probability of generating the symmetric group. *J. Combin. Theory Ser. A*, 52(1):148–153, 1989.

[11] Daniel Barath, Dmytro Mishkin, Ivan Eichhardt, Ilia Shipachev, and Jiri Matas. Efficient initial pose-graph generation for global sfm. *arXiv preprint arXiv:2011.11986*, 2020.

[12] Evangelos Bartzos, Ioannis Z Emiris, and Josef Schicho. On the multihomogeneous Bézout bound on the number of embeddings of minimally rigid graphs. *arXiv preprint arXiv:2005.14485*, 2020.

[13] Aravind Baskar and Sandipan Bandyopadhyay. An algorithm to compute the finite roots of large systems of polynomial equations arising in kinematic synthesis. *Mechanism and Machine Theory*, 133:493–513, 2019.

[14] Aravind Baskar, Chang Liue, Mark Plecnik, and Jonathan D. Hauenstein. Designing rotary linkages for polar motions. *To be presented at 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*.

[15] Daniel J Bates, Jonathan D Hauenstein, Andrew J Sommese, and Charles W Wampler. Bertini: Software for numerical algebraic geometry, 2006.

[16] Daniel J. Bates, Jonathan D. Hauenstein, Andrew J. Sommese, and Charles W. Wampler. *Numerically solving polynomial systems with Bertini*. SIAM, 2013.

[17] Carlos Beltrán and Anton Leykin. Robust certified numerical homotopy tracking. *Found. Comput. Math.*, 13(2):253–295, 2013.

[18] Carlos Beltrán and Luis Miguel Pardo. On Smale's 17th problem: a probabilistic positive solution, 2008.

[19] D. N. Bernstein. The number of roots of a system of equations. *Funkcional. Anal. i Priložen.*, 9(3):1–4, 1975.

[20] Nathan Bliss, Timothy Duff, Anton Leykin, and Jeff Sommars. Monodromy solver: sequential and parallel. In *Proceedings of the 2018 ACM International Symposium on Symbolic and Algebraic Computation*, pages 87–94, 2018.

[21] Madeline Brandt, Juliette Bruce, Taylor Brysiewicz, Robert Krone, and Elina Robeva. The degree of $\mathrm{SO}(n, \mathbb{C})$. 80:229–246, 2017.

[22] Paul Breiding, Türkü Özlüm Çelik, Timothy Duff, Alexander Heaton, Aida Maraj, Anna-Laura Sattelberger, Lorenzo Venturello, and Oğuzhan Yürük. Nonlinear algebra and applications. *arXiv preprint arXiv:2103.16300*, 2021.

[23] Paul Breiding and Sascha Timme. Homotopycontinuation. jl: A package for homotopy continuation in Julia. In *International Congress on Mathematical Software*, pages 458–465. Springer, 2018.

[24] Taylor Brysiewicz, Jose Israel Rodriguez, Frank Sottile, and Thomas Yahl. Solving decomposable sparse systems. *Numerical Algorithms*, 2021.

[25] Peter Bürgisser and Felipe Cucker. Solving polynomial equations in smoothed polynomial time and a near solution to Smale's 17th problem [extended abstract]. In *STOC'10—Proceedings of the 2010 ACM International Symposium on Theory of Computing*, pages 503–512. ACM, New York, 2010.

[26] Élie Cartan. *Leçons sur la géométrie projective complexe. La théorie des groupes finis et continus et la géométrie différentielle traitées par la méthode du repère mobile. Leçons sur la théorie des espaces à connexion projective*. Les Grands Classiques Gauthier-Villars. [Gauthier-Villars Great Classics]. Éditions Jacques Gabay, Sceaux, 1992. Reprint of the editions of 1931, 1937 and 1937.

[27] Justin Chen and Joe Kileel. Numerical implicitization for Macaulay2. *Journal of Software for Algebra and Geometry*, 9:55–65, 2019.

[28] Fernando Cukierman. Monodromy of projections. *Mat. Contemp.*, 16:9–30, 1999. 15th School of Algebra (Portuguese) (Canela, 1998).

[29] Bernard Deconinck and Mark van Hoeij. Computing Riemann matrices of algebraic curves. volume 152/153, pages 28–46. 2001. Advances in nonlinear mathematics and science.

[30] Persi Diaconis. *Group representations in probability and statistics*, volume 11 of *Institute of Mathematical Statistics Lecture Notes—Monograph Series*. Institute of Mathematical Statistics, Hayward, CA, 1988.

[31] John D. Dixon. The probability of generating the symmetric group. *Math. Z.*, 110:199–205, 1969.

[32] T Duff, C Hill, A Jensen, K Lee, A Leykin, and J Sommars. Monodromysolver: a Macaulay2 package for solving polynomial systems via homotopy continuation and monodromy, 2016.

[33] Timothy Duff, Cvetelina Hill, Anders Jensen, Kisun Lee, Anton Leykin, and Jeff Sommars. Solving polynomial systems via homotopy continuation and monodromy. *IMA Journal of Numerical Analysis*, 39(3):1421–1446, 2019.

[34] Timothy Duff, Kathlén Kohn, Anton Leykin, and Tomas Pajdla. PLMP-point-line minimal problems in complete multi-view visibility. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1675–1684, 2019.

[35] Timothy Duff, Kathlén Kohn, Anton Leykin, and Tomas Pajdla. PL1P—point-line minimal problems under partial visibility in three views. *Proceedings of the European Conference on Computer Vision*, 2020.

[36] Timothy Duff, Viktor Korotynskiy, Tomas Pajdla, and Margaret Regan. Galois/monodromy groups for decomposing minimal problems in 3d reconstruction. *arXiv preprint arXiv:2105.04460*, 2021.

[37] Timothy Duff and Michael Ruddy. Numerical equality tests for rational maps and signatures of curves. *Proceedings of the 2020 ACM International Symposium on Symbolic and Algebraic Computation*, 2020.

[38] Sean Eberhard and Stefan-Christoph Virchow. The probability of generating the symmetric group. *Combinatorica*, 39(2):273–288, 2019.

[39] Noam D. Elkies. The complex polynomials $P(x)$ with $\mathrm{Gal}(P(x) - t) \cong M_{23}$. 1:359–367, 2013.

[40] Ricardo Fabbri, Timothy Duff, Hongyi Fan, Margaret Regan, David da Costa de Pinho, Elias Tsigaridas, Charles Wampler, Jonathan Hauenstein, Benjamin Kimia, and Anton Leykin. TRPLP—Trifocal relative pose from lines at points (note: author order note alphabetical). *Proceedings of Computer Vision and Pattern Recognition*, 2020.

[41] Jean-Charles Faugere, Patrizia Gianni, Daniel Lazard, and Teo Mora. Efficient computation of zero-dimensional Gröbner bases by change of ordering. *Journal of Symbolic Computation*, 16(4):329–344, 1993.

[42] Mark Fels and Peter J. Olver. Moving coframes. I. A practical algorithm. *Acta Appl. Math.*, 51(2):161–213, 1998.

[43] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[44] André Galligo and Adrien Poteaux. Continuations and monodromy on random riemann surfaces. In *Proceedings of the 2009 Conference on Symbolic Numeric Computation*, SNC '09, page 115–124, New York, NY, USA, 2009. Association for Computing Machinery.

[45] The GAP Group. *GAP – Groups, Algorithms, and Programming, Version 4.11.1*, 2021.

[46] Daniel R. Grayson and Michael E. Stillman. Macaulay2, a software system for research in algebraic geometry. Available at http://www.math.uiuc.edu/Macaulay2/.

[47] Phillip Griffiths and Joseph Harris. *Principles of algebraic geometry*. Wiley Classics Library. John Wiley & Sons, Inc., New York, 1994. Reprint of the 1978 original.

[48] Elizabeth Gross, Heather A Harrington, Zvi Rosen, and Bernd Sturmfels. Algebraic systems biology: a case study for the wnt pathway. *Bulletin of mathematical biology*, 78(1):21–51, 2016.

[49] J. A. Grunert. Das pothenotische problem in erweiterter gestalt nebst über seine anwendungen in der geodäsie. *Grunerts Archiv für Mathematik und Physik*, 1:238–248, 1841.

[50] Jaime Gutierrez and David Sevilla. Building counterexamples to generalizations for rational functions of Ritt's decomposition theorem. *J. Algebra*, 303(2):655–667, 2006.

[51] Christian Häne, Lionel Heng, Gim Hee Lee, Friedrich Fraundorfer, Paul Furgale, Torsten Sattler, and Marc Pollefeys. 3D visual perception for self-driving cars using a multi-camera system: Calibration, mapping, localization, and obstacle detection. *Image and Vision Computing*, 68:14–27, 2017.

[52] Joe Harris. Galois groups of enumerative problems. *Duke Math. J.*, 46(4):685–724, 1979.

[53] Richard Hartley and Andrew Zisserman. Multiple view geometry in computer vision. 2004. *Google Scholar Google Scholar Digital Library Digital Library*, 2003.

[54] Allen Hatcher. *Algebraic topology*. Cambridge University Press, Cambridge, 2002.

[55] Jonathan D. Hauenstein, Ian Haywood, and Alan C. Liddell, Jr. An *a posteriori* certification algorithm for Newton homotopies. pages 248–255, 2014.

[56] Jonathan D Hauenstein, Anton Leykin, Jose Israel Rodriguez, and Frank Sottile. A numerical toolkit for multiprojective varieties. *to appear in Mathematics of Computation*, 2020.

[57] Jonathan D. Hauenstein, Luke Oeding, Giorgio Ottaviani, and Andrew J. Sommese. Homotopy techniques for tensor decomposition and perfect identifiability. *J. Reine Angew. Math.*, 753:1–22, 2019.

[58] Jonathan D. Hauenstein and Jose Israel Rodriguez. Multiprojective witness sets and a trace test. *Adv. Geom.*, 20(3):297–318, 2020.

[59] Jonathan D. Hauenstein, Jose Israel Rodriguez, and Frank Sottile. Numerical computation of Galois groups. *Found. Comput. Math.*, 18(4):867–890, 2018.

[60] Jonathan D Hauenstein and Andrew J Sommese. Witness sets of projections. *Applied Mathematics and Computation*, 217(7):3349–3354, 2010.

[61] Jonathan D Hauenstein and Andrew J Sommese. Membership tests for images of algebraic sets by linear projections. *Applied Mathematics and Computation*, 219(12):6809–6818, 2013.

[62] Jonathan D. Hauenstein, Andrew J. Sommese, and Charles W. Wampler. Regeneration homotopies for solving systems of polynomials. *Math. Comp.*, 80(273):345–377, 2011.

[63] Birkett Huber and Bernd Sturmfels. A polyhedral method for solving sparse polynomial systems. *Math. Comp.*, 64(212):1541–1555, 1995.

[64] Anders N. Jensen. Gfan, a software system for Gröbner fans and tropical varieties. Available at http://home.imf.au.dk/jensen/software/gfan/gfan.html.

[65] Camille Jordan. *Traité des substitutions et des équations algébriques*. Librairie Scientifique et Technique A. Blanchard, Paris, 1957. Nouveau tirage.

[66] B. Ya. Kazarnovskiĭ. Newton polyhedra and Bezout's formula for matrix functions of finite-dimensional representations. *Funktsional. Anal. i Prilozhen.*, 21(4):73–74, 1987.

[67] Gregor Kemper. Separating invariants. *J. Symbolic Comput.*, 44(9):1212–1222, 2009.

[68] Joe Kileel. Minimal problems for the calibrated trifocal variety. *SIAM Journal on Applied Algebra and Geometry*, 1(1):575–598, 2017.

[69] Irina A. Kogan, Michael Ruddy, and Cynthia Vinzant. Differential signatures of algebraic curves. *SIAM J. Appl. Algebra Geom.*, 4(1):185–226, 2020.

[70] A. G. Kouchnirenko. Polyèdres de Newton et nombres de Milnor. *Invent. Math.*, 32(1):1–31, 1976.

[71] Zuzana Kukelova, Martin Bujnak, and Tomas Pajdla. Automatic generator of minimal problem solvers. In *European Conference on Computer Vision*, pages 302–315. Springer, 2008.

[72] Zuzana Kukelova, Joe Kileel, Bernd Sturmfels, and Tomas Pajdla. A clever elimination strategy for efficient minimal solvers. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4912–4921, 2017.

[73] Pierre Lairez. A deterministic algorithm to compute approximate roots of polynomial systems in polynomial average time. *Found. Comput. Math.*, 17(5):1265–1292, 2017.

[74] Viktor Larsson and Kalle Åström. Uncovering symmetries in polynomial systems. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *Computer Vision – ECCV 2016*, pages 252–267. Springer International Publishing, 2016.

[75] Viktor Larsson, Kalle Astrom, and Magnus Oskarsson. Efficient solvers for minimal problems by syzygy-based reduction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 820–829, 2017.

[76] Viktor Larsson, Magnus Oskarsson, Kalle Astrom, Alge Wallis, Zuzana Kukelova, and Tomas Pajdla. Beyond grobner bases: Basis selection for minimal solvers. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3945–3954, 2018.

[77] Robert Lazarsfeld. *Positivity in algebraic geometry. I*, volume 48 of *Ergebnisse der Mathematik und ihrer Grenzgebiete. 3. Folge. A Series of Modern Surveys in Mathematics [Results in Mathematics and Related Areas. 3rd Series. A Series of Modern Surveys in Mathematics]*. Springer-Verlag, Berlin, 2004. Classical setting: line bundles and linear series.

[78] Anton Leykin, Jose Israel Rodriguez, and Frank Sottile. Trace test. *Arnold Math. J.*, 4(1):113–125, 2018.

[79] Anton Leykin and Frank Sottile. Galois groups of Schubert problems via homotopy computation. *Mathematics of Computation*, 78(267):1749–1765, 2009.

[80] T. Y. Li, Tim Sauer, and J. A. Yorke. The cheater's homotopy: an efficient procedure for solving systems of polynomial equations. *SIAM J. Numer. Anal.*, 26(5):1241–1251, 1989.

[81] Max Lieblich and Lucas Van Meter. Two Hilbert schemes in computer vision. *SIAM Journal on Applied Algebra and Geometry*, 4(2):297–321, 2020.

[82] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.

[83] Yi Ma, Stefano Soatto, Jana Kosecka, and S Shankar Sastry. *An invitation to 3-D vision: from images to geometric models*, volume 26. Springer Science & Business Media, 2012.

[84] Abraham Martín del Campo and Jose Israel Rodriguez. Critical points via monodromy and local methods. *J. Symbolic Comput.*, 79(part 3):559–574, 2017.

[85] Stephen Maybank. *Theory of reconstruction from image motion*, volume 28. Springer Science & Business Media, 2012.

[86] Mateusz Michałek and Bernd Sturmfels. *Invitation to nonlinear algebra*, volume 211. American Mathematical Soc., 2021.

[87] Alexander Morgan. *Solving polynomial systems using continuation for engineering and scientific problems*, volume 57 of *Classics in Applied Mathematics*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2009. Reprint of the 1987 original [MR1049872], Pages 304–534: computer programs section, also available as a separate file online.

[88] Alexander P. Morgan and Andrew J. Sommese. Coefficient-parameter polynomial continuation. *Appl. Math. Comput.*, 29(2, part II):123–160, 1989.

[89] David Nistér. An efficient solution to the five-point relative pose problem. *IEEE transactions on pattern analysis and machine intelligence*, 26(6):756–770, 2004.

[90] David Nistér and Frederik Schaffalitzky. Four points in two or three calibrated views: Theory and practice. *Int. J. Comput. Vis.*, 67(2):211–231, 2006.

[91] Dylan Peifer and Mahrud Sayrafi. FGLM: Groebner bases via the FGLM algorithm. Version 1.1.0. A *Macaulay2* package available at https://github.com/Macaulay2/M2/tree/master/M2/Macaulay2/packages.

[92] Mark M Plecnik and Ronald S Fearing. Finding only finite roots to large kinematic synthesis systems. *Journal of Mechanisms and Robotics*, 9(2), 2017.

[93] Srikumar Ramalingam, Sofien Bouaziz, and Peter Sturm. Pose estimation using both points and lines for geo-localization. In *2011 IEEE International Conference on Robotics and Automation*, pages 4716–4723. IEEE, 2011.

[94] Bernhard Riemann. Beiträge zur Theorie der durch die Gauss'sche Reihe $F(\alpha, \beta, \gamma)$ darstellbaren Functionen. In *Abhandlungen der Königlichen Gesellschaft der Wissenschaften zu Göttingen*, VII Math. Classe: A–22. 1857.

[95] J. F. Ritt. Prime and composite polynomials. *Trans. Amer. Math. Soc.*, 23(1):51–66, 1922.

[96] Joseph J. Rotman. *An introduction to the theory of groups*, volume 148 of *Graduate Texts in Mathematics*. Springer-Verlag, New York, fourth edition, 1995.

[97] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4104–4113, 2016.

[98] Jean-Pierre Serre. Topics in Galois theory. 1:xvi+120, 2008. With notes by Henri Darmon.

[99] Igor R. Shafarevich. *Basic algebraic geometry. 1*. Springer-Verlag, Berlin, second edition, 1994. Varieties in projective space, Translated from the 1988 Russian edition and with notes by Miles Reid.

[100] Noah Snavely, Steven M. Seitz, and Richard Szeliski. Modeling the world from internet photo collections. *International journal of computer vision*, 80(2):189–210, 2008.

[101] Andrew J. Sommese, Jan Verschelde, and Charles W. Wampler. Numerical decomposition of the solution sets of polynomial systems into irreducible components. *SIAM J. Numer. Anal.*, 38(6):2022–2046, 2001.

[102] Andrew J. Sommese, Jan Verschelde, and Charles W. Wampler. Symmetric functions applied to decomposing solution sets of polynomial systems. *SIAM J. Numer. Anal.*, 40(6):2026–2046 (2003), 2002.

[103] Andrew J Sommese, Charles W Wampler, et al. *The Numerical solution of systems of polynomials arising in engineering and science*. World Scientific, 2005.

[104] Sascha Timme. Mixed Precision Path Tracking for Polynomial Homotopy Continuation. *arXiv e-prints*, page arXiv:1902.02968, February 2019.

[105] Lucas Van Meter. *A Functorial Approach to Algebraic Vision*. PhD thesis, University of Washington, 2019.

[106] Jan Verschelde. Algorithm 795: PHCpack: A general-purpose solver for polynomial systems by homotopy continuation. *ACM Transactions on Mathematical Software (TOMS)*, 25(2):251–276, 1999.

[107] Prahlad Warszawski and Howard M Wiseman. Open quantum systems are harder to track than open classical systems. *Quantum*, 3:192, 2019.

[108] Juan Xu, Michael Burr, and Chee Yap. An approach for certifying homotopy continuation paths: univariate case. In *ISSAC'18—Proceedings of the 2018 ACM International*

Symposium on Symbolic and Algebraic Computation, pages 399–406. ACM, New York, 2018.